



Universidad Carlos III de Madrid

Departamento de Ingeniería de Sistemas y Automática

Trabajo de Fin de Grado:

**Implementación de un Traje con Sensores como Controlador MIDI para
Danza**

Autor:

Sandra Berrio Acosta

Grado en Ingeniería Electrónica Industrial y Automática

Supervisado por:

Javier Fernández de Gorostiza Luengo

TÍTULO: *Implementación de un Traje con Sensores como Controlador MIDI para Danza.*

AUTOR: *SANDRA BERRIO ACOSTA*

TUTOR: *JONATAN CRESPO HERRERO*

DIRECTOR: *JAVIER FERNÁNDEZ DE GOROSTIZA*

La defensa del presente Proyecto Fin de Carrera se realizó el día 4 de Julio de 2016; siendo calificada por el siguiente tribunal:

PRESIDENTE: *Marta Ruiz Llata*

SECRETARIO *Ricardo Valverde Gil*

VOCAL *Alberto Emmanuel Quintero Gamez*

Habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

Presidente

Secretario

Vocal

Agradecimientos

En primer lugar, quisiera dar las gracias a quien me dió la oportunidad de realizar este trabajo de fin de grado, mi tutor Javier Gorostiza. Me gustaría destacar tu optimismo e ilusión en todo lo que he ido consiguiendo, que me ha servido y servirá de motivación para esta y otras metas que me proponga. Muchas gracias por tu dedicación y consejos, ha sido un placer compartir esta etapa tan importante contigo.

A mi familia, en especial mis padres, Gabriely Carmina, que me han apoyado durante todo el proceso de elaboración del proyecto, demostrándome su confianza y ofreciéndome toda su ayuda. A mi hermana Carmen, quien me ha aguantado de buena forma cada día, intentando hacerme sentir mejor. Ha sido mi apoyo en todo momento, y por ello no puedo expresar toda mi gratitud.

A mis compañeros de carrera, los cuales han ayudado que la yo que soy en estos momentos sea la mejor yo que pueda haber. Mención especial a Almu, a quien conocí el primer día y quien ha estado conmigo desde entonces, haciéndome ver quien soy y mostrándome su infinita sabiduría. Y gracias a Álvaro, que ha aguantado más que nadie mis indecisiones y nervios, Noel y Alba por conseguir que sacarálo mejor de mí en las prácticas de la carrera.

No puedo olvidarme de la gente de mi Erasmus en Finlandia, en especial a Caterina, que me ha cuidado y me ha hecho ver el camino correcto cuando estaba perdida, y por todos los momentos que hemos pasado juntas. Muchas gracias por los ánimos en especial en esta recta final que lo he necesitado.

El verdadero progreso es el que pone la tecnología al alcance de todos.

Henry Ford

Resumen

Se ha querido plasmar en este documento la relación que puede existir entre arte y tecnología, en concreto con la danza, mostrando el lado más humano y sensible que nos ofrece la electrónica. Además, pretende informar sobre recientes plataformas con sensores inalámbricos, que se desarrollaron para actuaciones de danza interactiva.

El objetivo principal es la creación de un traje que incorpore todas las funciones a través del uso de ciertos sensores que permitan esta interacción entre danza - tecnología - espectador.

Por último, se presentará el diseño final del sistema de percepción de movimientos del bailarín, junto con algunos de los proyectos externos, que complementarían el objetivo final de crear una actuación presentada en convenciones internacionales.

Índice general

1. Introducción	1
1.1. Objetivos	3
1.2. Metodología y plan de trabajo	5
1.2.1. Diagrama de Gantt	6
1.3. Organización y estructura del documento	8
1.4. Estado del arte	10
2. Plataforma de trabajo	13
2.1. Introducción	13
2.2. La Plataforma Arduino	13
2.2.1. Hardware	14
2.2.2. Software	18
2.3. CoolTerm	20
2.4. Protocolo MIDI	20
2.5. Fritzing	23
2.6. OrCAD Capture CIS	23
3. Diseño del sistema	25
3.1. Elección de parámetros a detectar	25
3.2. Elementos del sistema de detección	28
3.2.1. Módulo de Bluetooth	28
3.2.2. Sensor IR SHARP GP2D120	29

3.2.3. Sensor ADXL3xx	29
3.2.4. Sensor ADXL345	30
3.2.5. Sensores de contacto	32
3.2.6. Conexión MIDI a un Arduino	33
4. Desarrollo e implementación	34
4.1. Comunicación inalámbrica	34
4.1.1. Introducción a XBee	34
4.1.2. Bluetooth	35
4.1.3. Comparación de tecnologías de comunicación	37
4.2. EZ - Link Adafruit	38
4.3. Sistema de detección de proximidad	39
4.3.1. Sensor de Infrarrojos SHARP IR GP2D120	41
4.4. Sistema de detección de posición	43
4.4.1. Acelerómetro ADXL 3xx	45
4.4.2. Acelerómetro ADXL 345	48
4.5. Sistema de detección de contacto	52
4.6. Integración sensorial en un sistema de control	55
4.6.1. Equipamiento	55
4.6.2. Descripción del funcionamiento	61
4.6.3. Montaje	62
5. Resultados y experimentos	65
6. Conclusiones	68
7. Posibles mejoras y trabajos futuros	70
7.1. Trabajos futuros	71
APÉNDICES	73
A. Presupuesto del Proyecto	73
A.0.1. Desglose por horas	73
A.0.2. Costes materiales	76

A.0.3. Costes de personal	77
A.0.4. Coste total final	77
B. Anexo I	78
B.1. <i>Wiring</i>	78
B.2. <i>Processing</i>	79
C. Anexo II	80

Lista de Figuras

1.1. Ballet del <i>Cascanueces</i> con sensores en Brooklyn	12
1.2. Par de bailarinas con sensores integrados	12
2.1. Detalle de Placa Arduino UNO	17
2.2. Arduino IDE	19
2.3. CoolTerm	20
2.4. Pantalla principal del software Frizting	23
2.5. OrCAD Capture CIS	24
3.1. Esquema general del sistema de percepción de movimientos del bailarín	27
3.2. Caja negra	27
3.3. Conexiones módulo BT a Arduino UNO	28
3.4. Conexión GP2D120	29
3.5. Esquema de conexión del Sistema de Dección de Proximidad	29
3.6. Esquema de conexión del ADXL3xx	30
3.7. Pines del ADXL345.	30
3.8. Conexiones ADXL345.	30
3.9. Esquema de un circuito simple para un ADX345	31
3.10. Esquema de un circuito doble para dos ADXL345	31
3.11. Conexión de un botón interruptor	32
3.12. Esquema de conexión del circuito con un único botón.	32
3.13. Esquema de conexión del circuito con 4 SWITCHES	32
3.14. Conexión conector hembra MIDI a Arduino	33

3.15. Conexiones entre ordenador y sintetizador	33
4.1. Módulo ZigBee (XBee)	35
4.2. Esquema de protocolo Bluetooth	36
4.3. Componentes EZ- link Adafruit	39
4.4. Ángulos de reflexión a distintas distancias	40
4.5. IR SHARP GP2D120	41
4.6. Ejemplo de gráfica para sensor IR SHARP	41
4.7. Medida de la gravedad del Acelerómetro	43
4.8. Gráfica de ángulo de inclinación frente a la aceleración de la gravedad	44
4.9. Acelerómetro ADXL3xx	45
4.10. Ejes de aceleración en relación con la sensibilidad.	46
4.11. Gráfica del ángulo de inclinación frente a la función de sensibilidad	46
4.12. Acelerómetro ADXL345	48
4.13. Respuesta de salida frente a la orientación de la g.	49
4.14. Gráfico del efecto rebote en el ON y OFF	52
4.15. Pila de 9 V	56
4.16. Arduino UNO + cable	56
4.17. ADXL 3xx	56
4.18. ADXL 345	56
4.19. EZ - Link módulo	57
4.20. Sensor IR SHARP	57
4.21. Láminas de Latón	58
4.24. Tela de neopreno	58
4.22. Protoboard	58
4.23. Baquelita para soldar	58
4.25. Muñequera y tobillera	59
4.26. Faja de neopreno	59
4.27. Cables	59
4.28. Aguja e hilo negro	59
4.29. Soldador y estaño	59
4.30. Multímetro	60

4.31. Ordenador portátil hp	60
4.32. Resitro de muestra en CoolTerm de un giroscopo	61
4.33. Esquema de conexiones de todo el sistema	63
4.34. Placa Arduino con módulo externo para conectores	64
5.1. Muñequera con sensor de movimiento ADXL345	66
5.2. Sensor ADXL 3xx	66
5.3. Muñequera con sensor ADXL3xx	67
5.4. Bilarín con el traje completo	67
7.1. Cámara Kinect	71
7.2. Imágen obtenida a través de un sensor Kinect	71
C.1. Actuación durante una de las presentaciones	80
C.2. Bailarines durante la actuación	81
C.3. Ejemplo de distintas funciones del trabajo	81
C.4. Traje interactuando con el público	82
C.5. Parte de la actuación del proyecto SynBody	82
C.6. Cartel Metabody	83
C.7. Cartel de Bellas Artes	83

Lista de Tablas

4.1. Comparación ZigBee y EZ-Link.	37
4.2. Principales características de algunos metales	53
A.1. Desglose de horas - fase 1	73
A.2. Desglose de horas - fase 2	74
A.3. Desglose de horas - fase 3	74
A.4. Desglose de horas - fase 4	75
A.5. Desglose total de horas	75
A.6. Desglose total de coste de materiales	76
A.7. Costes de personal	77
A.8. Coste total	77

Capítulo 1

Introducción

Actualmente la tecnología está presente básicamente en cualquier aspecto de nuestra vida cotidiana. Sin embargo, durante las últimas décadas el avance y uso de la tecnología en el arte se ha transformado en planteamientos cada vez más perfeccionados, su admirable desarrollo en el área del arte, han conseguido que haya un aumento en la incorporación de los avances tecnológicos, especialmente en la expresión plástica, como instalaciones interactivas, net art, robótica, etc. Esto ha sido debido a un incremento en la oferta en ferias, convenciones y festivales internacionales, como *Ars Electrónica* y *Transmediale* entre otros.

Este trabajo se hizo como propuesta para crear una unión entre danza y electrónica, dando la posibilidad de crear nuevas relaciones entre artista - obra - espectador, a través de un nuevo lenguaje transmitido por el cuerpo del bailarín y a través de la tecnología. Danza y tecnología son dos disciplinas que encajan perfectamente, siempre se está buscando caminos conjuntos para la obtención de nuevas formas de expresión.

Históricamente el siglo XX fue cuándo hubo más movimientos de rupturas hablando en la tendencia de lo escénico, más concretamente durante la década de los 60, que es cuando realmente empiezan a proliferar en el terreno de la experimentación con el movimiento y la tecnología. Desde entonces las posibilidades entorno a la danza y tecnología son infinitas, al ser complementarias provocan una innovación en el desarrollo de la tecnología y a su vez crea nuevas formas escénicas.

En la Universidad nos encontramos con un departamento donde se juega con la danza y la ingeniería electrónica y automática, que junto con investigadores externos permiten crear proyectos de artes escénicas con elementos tecnológicos. Con la colaboración de Eva Sanz, quien dirige *El Aula de las Artes* [15] junto con el bailarín y coordinador del aula de las artes, Alfredo Miralles y Javi F. Gorostiza, se ha podido desarrollar este proyecto para que en un futuro llegue implantarse en obras escénicas, permitiendo llegar al público este tipo de arte, cada vez más arraigado, y enseñándoles que ya no son simples observadores pasivos, si no que también forman parte de la puesta en escena.

Como primera aproximación hacia grandes actuaciones, se encuentra el proyecto *SynBody*, como se puede ver en su página [31], cuyo fundador ha sido Javier Aparicio, que en conjunto con este proyecto de traje se han ido experimentando hasta donde se puede llegar a innovar con la tecnología.

1.1. Objetivos

Cómo se comentó antes, el principal objetivo es la combinación de la electrónica y automática con el arte escénico, por ello el proyecto se ha llevado a cabo siguiendo una serie de objetivos, enumerados a continuación:

- El principal objetivo de este proyecto es el desarrollo de un traje con sensores que permita captar movimientos de un bailarín en escena, y transformar esas señales percibidas en mensajes MIDI, que puedan ser interpretados por una aplicación externa de generación sonora - musical.
- Permitir que la expresión del lenguaje que se quiere transmitir a través del baile, sea amplificada de tal manera que es espectador se sienta dentro del propio espectáculo.
- Componer un traje cuyas posibilidades sean el poder seguir siendo actualizado, y que sea fácil de llevar, simple de componer y cuyo presupuesto sea el mínimo.
- La posibilidad de lograr un nivel de exactitud y precisión mediante la tecnología que lleva a registrar pequeños cambios físicos.
- Componer en un futuro un pequeño espectáculo donde se muestren los resultados finales.
- Demostrar que la danza y tecnología van de la mano, mostrar a la gente que no hay tanta diferencia entre los dos campos, aunque uno sea del ámbito humanístico y el otro de la rama de la ingeniería.
- Enseñar que la electrónica no tiene que ser algo que carezca de sentimientos ya que puede ayudar a la creación de algo profundo y hermoso, al igual que es divertido y nos puede solucionar problemas del día a día.
- Integrar en un mismo traje sensores de movimiento, de posición y de contacto que se relacionen a través de una única plataforma de manera inalámbrica.
- Tiene que ser modular, es decir, debe permitir la integración de más funciones y sensores, o modificar los existentes, sin necesidad de grandes cambios globales en la plataforma.

- Todos los objetivos estan enmarcados dentro del trabajo del artista Javier Aparicio *Syn-Body*, y su investigación en la geración de señales sonoras y musicales através del movimiento.
- Aprender es el realizar un proyecto de ingeniería completo, en el que desarrollaría el papel como consultor tecnológico y creativo, que aunque en este caso en concreto esté centrado en la danza, sirva para cualquier tipo de tecnología existente.

1.2. Metodología y plan de trabajo

Para conseguir los objetivos propuestos para este Trabajo de Fin de Grado se ha diseñado y seguido el plan de trabajo que se detalla más abajo.

■ Estudio del Estado del Arte

Como en todo proyecto, es necesario tener una etapa previa de estudio y formación para adquirir los conocimientos requeridos para su desarrollo. En este caso en concreto, dicha formación se ha dirigido a la investigación de trabajos previos con los tipos de sensores usados para la detección de diversos parámetros, y también las posibilidades de detección comerciales. Además se ha realizado un amplio estudio de la plataforma Arduino para ver las opciones que ofrecerá al sistema diseñado.

■ Diseño y desarrollo del sistema de detección

La etapa principal del trabajo se centra en la implementación del sistema, que analizará los movimientos del bailarín. Se ha llevado a cabo un desarrollo del sistema en módulos, programando individualmente cada sensor y después implementándolo en el *hardware* necesario para su puesta en marcha.

■ Ajustes de los valores de los sensores

Una vez programado y desarrollado el esquema del circuito de cada sensor, se realizarán experimentos para verificar que el resultado es lo suficientemente riguroso. Así se podrán realizar cambios que mejoren su funcionamiento de manera aislada sin comprometer el funcionamiento del resto de sensores.

■ Montaje del sistema de detección final

Una vez que ya estén finalizados los subsistemas y habiendo comprobado su correcto funcionamiento, se adaptarán para formar el sistema final, pero manteniendo la independencia de cada sensor.

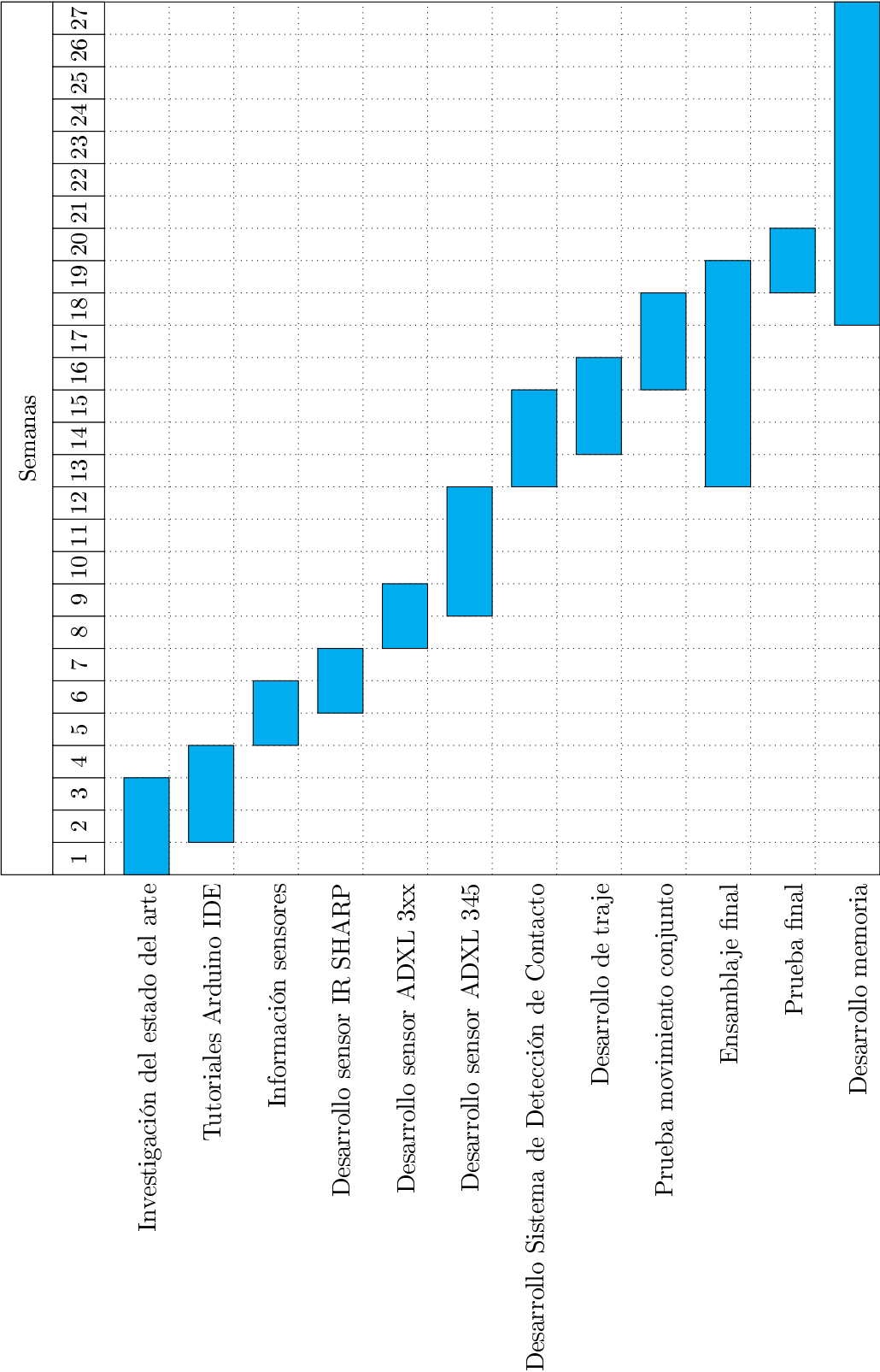
- **Ajustes finales del sistema de detección definitivo**

Una vez montado el sistema final, se harán las pruebas pertinentes para la última verificación de que el conjunto de todos los subsistemas funcionan como es debido. En caso de ser lo contrario cambiar los parámetros necesarios para obtener el resultado que se requiera.

- **Documentación y escritura de la memoria**

La parte final de este proyecto se ha empleado para documentar y explicar el trabajo realizado durante cada una de las etapas, registrando toda la información recopilada y necesaria para poder comprenderlo y desarrollarlo sin necesidad de material adicional.

1.2.1. Diagrama de Gantt



1.3. Organización y estructura del documento

El documento se divide en cuatro secciones:

- **Capítulo 1: Introducción**

En este capítulo se exponen el porqué se ha decidido implementar y desarrollar este proyecto, junto con los objetivos que lo inspiran, así como la metodología y el plan de trabajo llevado a cabo.

- **Capítulo 2: Plataforma de trabajo**

Aquí se presenta una panorámica actual de la plataforma Arduino sobre la que se pretende diseñar el detector, al igual que todos los complementos que tiene, como el software que facilita para la programación de las PCB (*Printed Board Circuit*). También se describen otras herramientas utilizadas para el diseño e implementación de proyecto.

- **Capítulo 3: Diseño del sistema de detección**

Aquí se desarrolla el proceso de diseño y puesta en marcha del sistema. Se exponen algunos aspectos a tener en cuenta en el diseño en lo referente a qué se va a detectar y qué sensores usar para esa detección en concreto. Se explica también la red inalámbrica utilizada y una comparación con otros dispositivos inalámbricos.

- **Capítulo 4: Desarrollo e implementación**

En este capítulo se desarrollan todas las características de los sensores que se han diseñado en el apartado, como una comparación con otros dispositivos, explicando el porqué de su uso.

- **Capítulo 5: Resultados y experimentos**

Este capítulo resume el trabajo que se ha llevado a cabo en este proyecto, observando que los resultados nos han aportado a la hora de sus pruebas y experimentos.

- **Capítulo 6: Conclusiones**

Se va a verificar si se han llevado a cabo todos los objetivos que se propusieron en el apartado de objetivos.

- **Capítulo 7: Posibles mejoras y trabajos futuros**

Se hará referencia a mejoras que se pueden implementar en el sistema de detección desarrollado, como otros métodos para complementarlo. Se hará referencia también a líneas futuras para este sistema.

- **Anexos: Presupuesto del trabajo**

Se incluye en detalle el tiempo que se ha utilizado para llevar a cabo este proyecto, al igual que los materiales utilizados, haciendo una valoración aproximada del coste real que tendría.

1.4. Estado del arte

Dentro de los proyectos en los que están relacionados danza y tecnología, caben destacar formatos que ya se consideran clásicos, pero que siguen evolucionando constantemente. La posibilidad que existe en las distintas interacciones con varios formatos, potencia la relación entre los lenguajes artísticos. Básicamente estas primeras interacciones se llevaron a cabo con el uso del video, el uso de un ordenador y la combinación de ambos. En concreto hay músicos que utilizan un programa llamado *Max*, siendo un programa que permite al compositor pautar distintas cláusulas, como es Jorge Sad, que es un compositor que viene investigando el movimiento de los músicos al tocar, componiendo obras electroacústicas aplicadas a multimedia y cine [23].

En el siglo XX Rudolf von Laban, teórico y bailarín, desarrolló un sistema de notación y análisis del movimiento que fue el predecesor que sentó los pilares fundamentales para la danza moderna, y como dijo en uno de sus libros:

Ibid.: p. 4. While there are numerous terms used to describe this practice, I will rely on three which I feel are the most precise. The first, dance for the camera is an inclusive term that refers to any and all dance created specifically for the camera, either in the medium of film or video. The second, cine-dance is a term that refers to work made in the medium of film, and video dance which refers to work made for the camera using the contemporary medium and practices of video technology. All three terms refer to the art of creating a choreography for the camera, to be viewed as a fully formed, autonomous work of art, ultimately, either on a film screen or television.

Y Cunningham, bailarín y coreógrafo americano, fue quien propuso el valorar el movimiento por sí mismo [17]. Hasta ahora se han desarrollado programas específicos para danza, con la posibilidad de aplicación a tiempo real en un espectáculo. Se trata de un mapeo del movimiento a través de cámaras conectadas a un ordenador, y que a través de ciertos programas provocará determinadas respuestas sonoras de video o lumínicas. El aspecto más desarrollado es la relación movimientos - estímulo sonoro. La coreógrafa Silvina Szperling, organizó el primer festival de videodanza, donde el objetivo es el intercambio de conocimientos, talleres de formación y estimular la producción [24].

Se puede decir con certeza que hasta hace relativamente poco el método más común de combinar danza y tecnología es con una cámara que leyese los movimientos, tenemos como ejemplos de varios artistas como la prepuesta de Garry Hill [25], que propone una acción la cual es registrada desde diversos puntos de su cuerpo, acompañando esos gestos en el movimiento; o Tamara Cubas, que integra movimiento y representación mediante dispositivos electrónicos que traducen su movimiento en imagen y sonido [26]. Sin embargo esto se centra únicamente en la imagen del movimiento, cuando es en el cuerpo donde sucede la acción.

Sabiendo a la velocidad que avanza la tecnología y lo rápido que nos llega a nuestras vidas, más y más compositores se interesan por captar el movimiento como tal. Existe una salida muy arraigada que es la de la creación de trajes instrumento, como nos enseña el artista Alex Posada [18], cuya idea es la de generar experiencias acústicas y visuales a la vez, ha trabajado a su vez con grandes músicos como *Chicks on Speed*, desarrollando trajes con sensores para sus actuaciones, una alfombra tejida con hilo conductor que funciona como un theremin gigante [19].

Es por eso que se están implementando sensores que les ayudan a visualizar los recorridos que hacen los bailarines durante un periodo de tiempo, para luego poder estudiar los datos obtenidos. El Laboratorio de Informática Musical de la Universidad de Guanajuato hizo un estudio titulado *SICIB: Sistema Interactivo de Composición e Improvisación Musical para Bailarines*, su función es integrar/acoplar coreografía y música permitiendo realizar obras musicales a través de los movimientos corporales [27]. O simplemente añaden acelerómetros para crear un efecto visual en el tutú de una bailarina de ballet cuando hace cierto movimiento, acentuando la historia que se está contando; o introducir de *atrezzo* unos guantes para el bailarín con galgas extensiométricas, que miden la presión que ejerce su mano [2]. Esto nos indica que las grandes compañías de baile están incluyendo muchos de estos dispositivos en sus actuaciones para añadir un valor extra en el espectáculo. Cada vez más *wearable technology*, o tecnología para llevar, se integra en la ropa de los bailarines, como por ejemplo unas bailarinas con unos sensores de presión en las suelas diseñado por Lesia Trubat González, como aparece en la imagen 1.2.

Hasta ahora no hay muchas obras conocidas con esta tecnología, pero cabe mencionar algunas que han tenido éxito, como el *El cascanueces* [28] en el ballet de Brookling [1.1], *Dust* una ópera de Robert Ashley [30], entre otros espectáculos. Esto nos demuestra que la incorporación de tecnología nos lleva más allá de la danza contemporánea, creciendo así el círculo de *techperformances* (actuaciones tecnológicas). En un futuro esta unión de arte y tecnología irá aumentando, a la vez que los métodos que se usen irán avanzando llegando a tener una firme unión entre tecnología y expresión del arte.



Figura 1.1: Ballet del *Cascanueces* con sensores en Brooklyn



Figura 1.2: Par de bailarinas con sensores integrados

Plataforma de trabajo

2.1. Introducción

Para el diseño de este sistema de detección se han utilizado una serie de herramientas y programas necesarios para el almacenamiento de los datos obtenidos y su posterior lectura. Dichos programas y herramientas se describen a continuación.

2.2. La Plataforma Arduino

Arduino es una plataforma *open-source* (código abierto) basado en un *hardware* y *software* flexibles y con un uso sencillo y básico para el usuario. Está basado para el diseño y desarrollo de prototipos interactivos autónomos, recibiendo *inputs* (entradas) desde sus puertos analógicos y digitales, y emitiendo *outputs* (salidas) desde los mismos, para con ellas intervenir en su entorno a través de distintos mecanismos.

El microcontrolador utiliza un lenguaje propio llamado *Arduino Programming Language*, (basado en el *Wiring*¹) y en el *Arduino Software* o *Arduino integrated Development Environment* (basado en el *Processing*²).

El desarrollo de proyectos en Arduino permiten autonomía y una comunicación con el software que se ejecuta en el ordenador. Por otro lado, las placas pueden ser ensambladas manualmente o adquirirlas ya preensambladas. Hay muchos diseños de hardware en open-source para ser utilizados o modificarlos para adaptarlos a las necesidades de cada uno.

¹Ver breve explicación en el Anexo I - Wiring

²Ver breve explicación en el Anexo I - Processing

Existen otras muchas plataformas de microcontroladores, Raspberri Pi, BQ, etc. entre otras. Sin embargo Arduino ofrece ciertas ventajas haciendo que sea de las más demandadas en varios campos:

- Se trata de un plataforma de bajo coste en comparación con el resto de microcontroladores.
- Hace uso de un software multiplataforma, es decir, puede ser utilizado tanto en Windows como Macintosh OSX/Linux.
- El entorno de programación es simple y conciso, en especial para usuarios principiantes, sin olvidar que es lo suficientemente flexible como para usuarios con conocimientos avanzados.
- Tiene un software adaptable. En el caso de programadores experimentados como la posibilidad de instalar librerías C++.
- Cuenta con un Hardware adaptable también. En un principio hacían uso de microprocesadores Atmel, pero en la actualidad existen otros modelos. A parte, los planos para módulos están publicados bajo licencia Creative Commons, lo que permite su continua mejora.

2.2.1. Hardware

El diseño de Hardware de Arduino consta de una placa con un microcontrolador y unos puertos de entrada/salida. En el mercado hay gran diversidad de placas electrónicas con distintas características, dando un mayor rango de posibilidades al usuario siguiendo sus necesidades. Además de los módulos y accesorios que permiten la mejora de estas placas básicas. Se ha de mencionar, que Arduino ha creado placas distintas teniendo en cuenta la localización fuera de EEUU o dentro de EEUU, las primeras las nombran como GENUINO y a las segundas como ARDUINO. Sin embargo las características de las placas no varían en absoluto. A continuación se presentan los posibles modelos disponibles dentro de la marca Arduino.

■ Placas E/S

Dependiendo de el nivel en el que quieras desarrollar tu prototipo, Arduino diferencia las placas en varios niveles, teniendo en cuenta el gran número de placas existentes, se presentará una breve descripción de las más conocidas y utilizadas habitualmente.

- **Nivel Básico**

ARDUINO UNO

Fue la primera de las series con USB. Esta basada en el ATmega328P. Se conecta al ordenador por un cable USB estándar y tiene todo lo necesario para programar y usar la placa. Es la más utilizada por los usuarios, ya que es robusta y en caso de romper el chip, se puede reponer sin mayor dificultad y coste.

ARDUINO PRO

Es un microcontrolador basado en el ATmega168 or ATmega328. En este caso, la placa está pensada para proyectos semipermanentes. Viene en versiones tanto de 3.3V/8 MHz como de 5V/16MHz. Cuenta con 14 pines digitales de E/S (de los cuales 6 pueden utilizarse para salidas PWM), 6 entradas analógicas, un conector de alimentación de batería, un interruptor de encendido, un botón de reset, y agujeros para el montaje de un conector de alimentación, un cabezal ICSP, y cabezales para pines. La propia placa te permite conectar varios tipos de conectores, o soldar directamente sobre la placa.

- **Funciones Avanzadas**

ARDUINO MEGA

Diseñada para proyectos más complejos, y se recomienda para el uso de impresoras 3D. Tiene 54 E/S digitales, de los cuales 15 pueden ser usados como PWM salidas, 16 entradas digitales, 4 UARTS (puertos serie), un oscilador de cristal de 16MHz, un botón de reset y una cabecera ICP. La placa microcontroladora está basada en el ATmega2560. Es compatible con escudos de la UNO y Duemilove.

ARDUINO DUE

Placa que cuenta con 54 pines digitales de entrada/salida (12 pueden usarse para salidas PWM), 12 entradas analógicas, 4 UART, un reloj de 84 MHz, una conexión USB OTG, 2 DAC, 2TWI, un conector de alimentación, un cabezal de SPI, un header JTAG, un botón de reinicio y un botón de borrado. Es la única placa cuya alimentación es de 3.3V, y debe ser el voltaje máximo para los pines de entrada y salida.

- **Cosas de Internet**

ARDUINO YÚN

Es una placa basada en el ATmega32u4 y el Atheros AR9331. El procesador *Atheros* es compatible con Linux, el cual se basa en *OpenWrt*³ y *OpenWrt-Yun*. Se le ha incorporado Ethernet y soporte WiFi, ranura para tarjeta micro-SD y un puerto USB-A. Consta de 20 entradas digitales/pines de salida (7 puede utilizarse para salidas PWM y 12 como entradas analógicas), un oscilador de cristal de 16MHz, una conexión micro USB, una cabecera ICSP, y 3 botones de reset.

- **Ponibles**

LILYPAD ARDUINO USB

Es una placa perfecta para e-textiles y proyectos ponibles. Se cose a la tela junto con la batería, sensores y actuadores a través de un hilo conductor. La placa está basada en el ATmega32u4. Tiene 9 pines digitales de E/S (de los que pueden usarse 4 para salidas PWM), 4 entradas analógicas, un resonador 8MHz, una conexión micro USB, un conector JST para una batería de 3.7 V LiPo, y un botón de reset. Requiere sólo de conexión al ordenador o a una alimentación con batería para empezar a usarla. Como su propio nombre indica, la placa incorpora la comunicación.

Teniendo en cuenta todas las posibles placas con sus diferentes características y sabiendo la embergadura del proyecto a realizar, se optó por el uso de una Arduino UNO, ya que consta de el suficiente número de entradas analógicas y digitales para los sensores usados. Al igual que tiene una capacidad interna lo suficientemente potente como para procesar todos los datos de una forma rápida y concisa. El modelo con sus respectivas entradas y salidas, está detallado en la Figura 2.1.

³Es un firmware basada en Linux, y que nos permite aprovechar las características técnicas de un router.

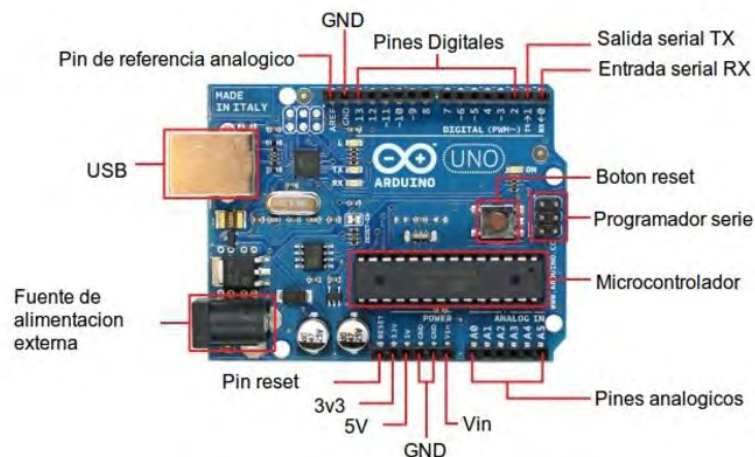


Figura 2.1: Detalle de Placa Arduino UNO

■ Módulos

Como en el caso de las placas, dependiendo del lugar de venta hay distintos modelos de módulos para ARDUINO y GENUINO, habiendo más variedad para ARDUINO.

- *Arduino PRO MINI*

Esta placa está basada en una ATmega328. Se usa para instalaciones semipermanentes, viene con la posibilidad de conectarse a distintos tipos de conectores o bien la posibilidad de soldar sobre ella misma. Es compatible con Arduino MINI. Consta de 14 puertos de E/S digitales, 6 de los cuales pueden ser usados como salidas PWM. 6 entradas analógicas, un resonador incluido en la placa y un boton de reset.

- *Arduino MICRO*

Es la placa más pequeña, basada en ATmega32U4. Lleva incluido un micro USB como conexión. Con 20 puertos de entrada y salida, 7 de ellos con posibilidad de ser salida PWM y 12 entradas digitales. Un oscilador de cristal de 16MHz y una cabecera ICSP.

- *Arduino NANO*

Una placa basada en ATmega328, y tiene un funcionamiento similar al Arduino Duemilanove. Trabaja con un USB Mini-B.

■ **Carcásas:**

- *Arduino Motor Shield:*

Se trata de un puente conductor dual diseñado para conducir cargas inductivas como relees, solenoides, etc. Te permite controlar la velocidad y la dirección de cada uno de los motores independientemente, entre otras posibilidades.

- *Arduino Proto Shield*

Nos permite diseñar fácilmente circuitos. Te permite soldar en ella para crear los prototipos o conectar directamente los circuitos integrados en ella.

- *Arduino Ethernet Shield*

Permite la conexión a internet de la placa de arduino. Se conecta a la red mediante un cable RJ45 y siguiendo unas instrucciones básicas para controlarlo por internet.

- *WiFi Shield 101*

Permite la conexión a internet de forma inalámbrica, sin necesidad de contraseña ni ninguna configuración adicional.

2.2.2. Software

A parte del hardware, las placas Arduino necesitan que se carguen en sus microcontroladores los *sketches*⁴, para ello se debe usar el *freeware*⁵ de Arduino en su página web. En el caso de que la placa se conecte a través del puerto serie, para tener un correcto funcionamiento se han de instalar los drivers USB para el chip FTDI de la placa.

■ **Arduino IDE**

El IDE usa un compilador de lenguaje C modificado para crear, traducir y transmitir el código a la placa microcontroladora, también ofrece a los usuarios más principiantes simplicidad a la vez que permite el soporte para la programación avanzada en C. El entorno de desarrollo está escrito en Java y basado en *Processing*, AVR-GCC⁶ y otro software de código abierto.

⁴Códigos programables para las aplicaciones de Arduino

⁵Puede ser descargado sin coste alguno, pudiendo aportar la voluntad para ayudar a los desarrolladores

⁶Se refiere a una librería en C de alta calidad para su uso con GCC en los controladores ATmel AVR, descrita por el ANSI X3.159-1989

Arduino IDE está compuesto por un editor de texto dónde se escribe el código con los comandos, una consola dónde aparecerá la salida de tu programa y una barra de herramientas con funciones. Cuenta también con un espacio donde te aparecen mensajes de errores e informativos, siendo más fácil para el usuario verificar su código rápida y sencillamente. El propio programa viene con librerías integradas, que te permiten usar comandos que únicamente con el lenguaje en Arduino no te hubiera permitido realizar. También son muy útiles ya que te ayudan a recrear funciones complejas en un número reducido de líneas de comandos.

Como se puede observar en la figura 2.2, aparecen dos funciones principales nada más abrir un nuevo proyecto. La primera es el *setup*, ahí se introduce el número de puerto serie de tu Arduino para que pueda ser leído por tu ordenador, y se añaden aquellas funciones que se tengan que llevar a cabo una sola vez.

La función de *loop*, como la misma palabra indica en inglés, bucle, se incluyen aquellas que tengan que leer/escribir valores un número de veces, ya sea determinado o indeterminado. En cuanto a las variables se suelen declarar al principio junto con las librerías que se vayan a usar.

En la esquina superior izquierda aparece una lupa, que se corresponde a la consola. Los demás botones son para la verificación del código y tras este paso el botón de subir el programa a la placa de Arduino.



Figura 2.2: Arduino IDE

2.3. CoolTerm

Coolterm [2.3] es un programa *freeware* fácil de usar para comunicaciones entre cualquier hardware conectado al puerto serie.

Es una simple aplicación puerto serie para quien necesite el intercambio de datos con el hardware conectado al puerto serie, como podría ser una placa microprocesadora, un servo controlador, etc.



Figura 2.3: CoolTerm

Con este programa podemos exponer los datos recibidos de nuestro puerto serie en formato texto o formato hexadecimal según lo configures. Capturar esa información en forma de ficheros de texto y después enviarlos. La última versión es la 1.4.5 del día 14 de Febrero de 2015.

2.4. Protocolo MIDI

MIDI corresponde a Interfaz Digital de Instrumentos Digitales. Fue desarrollado por Dave Smith. Está basado en un protocolo de comunicación serial estándar que permite que los ordenadores, sintetizadores y otros dispositivos musicales electrónicos comunicarse y compartir información para la generación de sonidos. Permite que múltiples instrumentos sean tocados desde un solo controlador.

La base de esta comunicación es el byte. Cada comando MIDI tiene una secuencia de bytes específica. El sistema MIDI solo puede transmitir señales en un sentido. Es decir la dirección va del dispositivo maestro, que es el que genera la información, al dispositivo esclavo, que la recibe. Lleva mensajes de eventos que especifican notación musical, tono y velocidad; señales de control para parámetros como son la dinámica, el vibrato, paneo, cues y señales de reloj que establecen y sincronizan el tiempo entre varios dispositivos.

Los dispositivos MIDI se agrupan generalmente en dos clases:

- Controladores, es decir dispositivos que generan señales MIDI en base a acciones humanas.
- Sintetizadores, incluyendo samplers, secuenciadores, etc.

MIDI es un protocolo serie que funciona a 31.250 baudios por segundo, en forma de puerto serie y así enviar los datos a esa velocidad. Los bytes en MIDI se dividen en dos grupos:

- Bytes de comando, siendo siempre 128 o mayor, 0x80 a 0xFF en hexadecimal. Estos son los que se encargan de la desactivación de una nota, *pitch bend* entre otros.
- Bytes de datos, son menores que 127, y comprenden desde 0x00 a 0x7F, se centran en el tono de la nota, la velocidad o la intensidad de esa nota, también controla la cantidad de variación del tono.

Los mensajes MIDI comprenden un byte de estado (bit 7 = 1), seguido de los bytes de datos (bit 7 = 0). Los mensajes se dividen en dos categorías principales:

- Mensajes de canal contienen un canal de cuatro bits codificado dentro del byte de estado, que manda el mensaje específicamente a uno de los otros 16 canales.
- Mensajes de sistemas, son mensajes sin codificar con números de canales, y están divididos en tres tipos principales: Sistema Común, Sistema a Tiempo Real y Sistema Exclusivo.

El código base para conectar un microcontrolador Arduino a un sistema MIDI sería el siguiente:

```
void setup ( ) {  
  // velocidad de transmisión: de serie. empiezan (3125 );  
}  
  
void loop ( ) {  
  // notas de juego de F # -0 (0x1E) a F # -5 (0x5A):  
    for(int nota = 0x1E; cuenta<0x5A; tenga en cuenta ++ ) {  
  // nota sobre el canal 1(0x90),  
  // un poco de valor de la //nota(nota), la velocidad media(0x45):  
    noteOn ( 0x90 , nota , 0x45 );  
    retardo ( 100 );  
  // nota sobre el canal 1 ( 0x90),  
  // un poco de valor de la nota(nota),  
  // la velocidad del silencio (0x00):  
    noteOn ( 0x90 , nota , 0x00 );  
    retardo ( 100 );  
    }  
}  
  
// reproduce una nota MIDI. No comprueba para ver que  
// cmd es mayor que 127, o que los valores de los datos son menos
```

Existen ciertos comandos que están estandarizados, entre ellos se encuentran [20]:

- *noteOn*, se designa como 0x90, el 9 nos indica que estamos en este comando y el otro dígito a que canal nos estamos refiriendo. Este comando se envía cuando una nota se deja de pulsar.
- *noteOff*, se designa como 0x80. Este comando se envía cuando una nota pulsa, caso contrario al anterior.
- *ControlChange*, aparece cuando un valor del controlador varía. Estos cambios pueden ser dados por pedales y palancas. Su comando es el 0xC0.

2.5. Fritzing

Es una herramienta CAD *freeware* cuyo objetivo es apoyar a los usuarios a diseñar productos reales a partir de prototipos.

Este software permite al diseñador documentar su prototipo basado en Arduino y crear un PCB de diseño para la fabricación, como se muestra en la figura 2.4 Además te permite subir a un sitio web asociado todos los proyectos y compartirlos con otros usuarios, para que haya una línea abierta de intercambio de conocimientos.

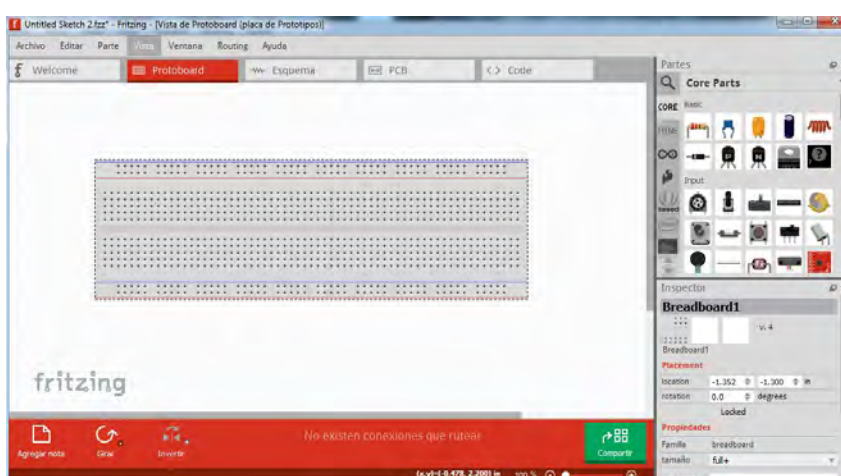


Figura 2.4: Pantalla principal del software Fritzing

Fritzing te permite la opción de trabajar sobre la automatización de diseño electrónico, ideada para los no ingenieros. Otra ventana te permite ver el código y modificarlo y subirlo directamente a un dispositivo Arduino. En este proyecto se ha usado la version 0.9.2b, que salió el 3 de Abril de 2015.

2.6. OrCAD Capture CIS

Es una herramienta de la empresa Candence, como se observa en su logo que aparece en la figura 2.5, para la captura de esquemáticos y diseño de circuitos. Con OrCAD Capture CIS (Component Information System) cualquiera puede acceder a información de distintos componentes eléctricos y electrónicos para crear el diseño final deseado. A partir de este esquemático se obtienen "netlists" de las que se usan para simulaciones con distintos parámetros el diseño de la PCB (*Printed Circuit Protoboard*).

Tiene distintas extensiones dependiendo de para lo que se vaya a utilizar, algunas de estas son:

- OrCAD Layout, para el diseño físico de la PCB, donde te incluyen diferentes huellas de componentes electrónicos del mercado.
- OrCAD PSpice para las simulaciones del esquemáticos del circuito. Dependiendo de las especificaciones dadas, puede crear análisis en DC, AC y tansitorias.

En este proyecto se ha utilizado principalmente el OrCAD Capture para el esquemático del circuito y el Layout para la PCB donde irán conectados los conectores para los sensores y entradas analógicas y digitales de la placa Arduino. Las versiones utilizadas son la 15.6 disponibles en las aulas informáticas de la universidad.



Figura 2.5: OrCAD Capture CIS

Diseño del sistema

El funcionamiento del sistema es el siguiente: el sistema recibe la información del cuerpo a través de los sensores incluidos en el traje. Una vez que se han leído los datos obtenidos, se procesa la información y a través de señales *MIDI*⁷ se traducen al lenguaje para controlar la información del entorno.

El proyecto comienza con la extracción de los puntos de los gestos y posición del cuerpo del bailarín. Para obtener esos puntos se hace uso de distintos sensores colocados en puntos diferentes del cuerpo. En los siguientes puntos se hará una descripción detallada donde se explica el porqué del uso de los sensores utilizados.

3.1. Elección de parámetros a detectar

En esta sección se va explicar la información más relevante con el objetivo de realizar la selección de los puntos a detectar por el sistema.

El principal objetivo del proyecto es capturar las distintas posiciones que realiza el bailarín a través de los movimientos, para luego convertirlas en una señal que modificará el entorno de este.

El número de sensores en el traje depende principalmente de la placa microcontroladora utilizada, ya que a mayor número de entradas analógicas, mayor posibilidad de incluir más sensores. Por lo que al haber utilizado una placa ARDUINO UNO, el número de sensores en el traje es de 4, sin incluir aquellos de los cuales tienen entradas digitales.

⁷ *Musical Instrument Digital Interface*

Para tener el máximo número de puntos obtenidos, los sensores se han localizado en sitios específicos en función del campo de trabajo de cada sensor.

También se ha de tener en cuenta el movimiento del bailarín, por lo que previamente se ha estudiado cuales son los pasos básicos más comunes en cualquier tipo de danza, para así encontrar puntos comunes que nos ayuden para obtener el lugar mejor posicionado de los sensores.

Así se llegó a la conclusión de hacer una división del cuerpo, para abarcar el mayor rango posible. Para ello el cuerpo se ha dividido en tres partes:

- Parte inferior, compuesto por las pierdas y tobillos.
- Tronco, compuesto por la cintura, pecho y espalda.
- Parte superior, formado por los brazos y muñecas.

Finalmente, habiendo hecho estas distinciones de lugar del cuerpo y posibles movimientos, el diseño del sistema cuenta con un detector de proximidad y tres detectores de posición. Para este proyecto se optó por la opción del uso del módulo EZ-Link del fabricante Adafruit para establecer la comunicación entre el sistema de detección y la estación de control.

A partir de la información de estos sensores, se obtienen los siguientes elementos coreográficos:

- *Curvatura y torsión de movimientos.* Un aspecto fundamental es la curvatura y torsión de los movimientos. Hasta ahora, aquellos que han hecho el estudio de esta parte de la danza se han apoyado en el teorema de *Frenet - Serrett*.⁸
- *Posición física del bailarín.* Se puede utilizar para caracterizar movimientos realizados en diferentes zonas.
- *Posición de algún elemento del cuerpo del bailarín.* Nos permite determinar si el bailarín está con las manos arriba, en el suelo, agachado, etc.
- *Velocidad y aceleración de desplazamiento.* A partir de las posiciones del sensor en un determinado tiempo, es posible establecer la aceleración y velocidad del desplazamiento de este.
- *Cambios repentinos.* Teniendo en cuenta la posición del sensor en el tiempo es posible determinar cambios repentinos, tales como giros, saltos y caídas.

⁸El cálculo de la curvatura nos permite caracterizar los movimientos de los bailarines independientemente de su orientación.

Como se puede observar en la figura 3.1, se muestra la idea principal del diseño de este proyecto, sensores que lo forman como la recopilación de datos, junto con la salida que se obtendrá.



Figura 3.1: Esquema general del sistema de percepción de movimientos del bailarín

En cuanto al traje físico, con el objetivo de esconder y mantener la placa lo más asegurada posible, se ha decidido diseñar una caja negra, como se observa en la imagen 3.2 en la que los sensores quedan a la vista, para que puedan realizar su detección. Sin embargo, las conexiones a la placa Arduino, el módulo de de bluetooth EZ - Link y la batería, junto con otros componentes estarán en el interior de la caja.

La caja está hecha de cartón, alrededor está cubierta de neopreno. En vez de estar cosida directamente a la faja, tiene cosidas en ambos lados velcro para poder sacar la placa cuantas veces sea necesario únicamente retirando el neopreno.



Figura 3.2: Caja negra

3.2. Elementos del sistema de detección

Como se ha enumerado en el apartado anterior, los sensores que formarán parte del sistema, este apartado explicará como se han de colocar para llevar a cabo el proyecto según la imagen 3.1.

3.2.1. Módulo de Bluetooth

El modelo de bluetooth es el Adrafruit EZ - Link, cuyas propiedades de describirán en el siguiente capítulo.

Un dato a destacar, es el hecho que este módulo no necesita de ningún código en Arduino para emparejarlo con nuestra placa y ordenador.

Con conectar de la forma que se muestra a continuación, además de a una fuente de voltaje de 9V, el ordenador lo va a reconocer como si fuese un dispositivo bluetooth cualquiera, es decir, una vez que se ha emparejado el bluetooth con el ordenador, sin necesidad de ningún comando en el IDE de Arduino empezará a recibir y enviar datos automáticamente.

Las conexiones del bluetooth con Arduino se muestran en la imagen 3.3.

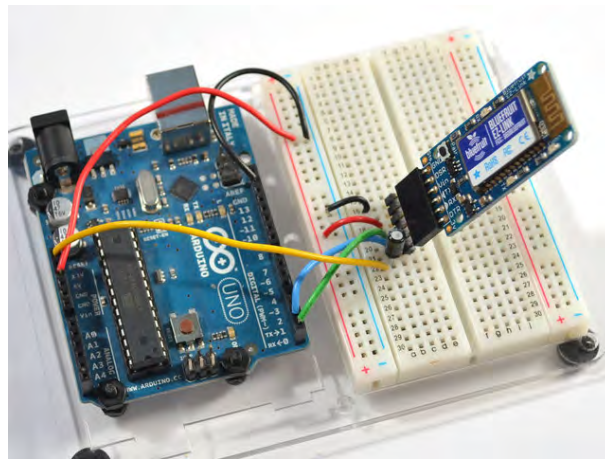


Figura 3.3: Conexiones módulo BT a Arduino UNO

3.2.2. Sensor IR SHARP GP2D120

■ Diagrama de conexión

Para realizar la conexión, este utiliza un conector llamado el conector de terminales sin soldadura (JST - *Japan Solderless Terminal*). Estos conectores tienen tres cables, tierra, Vcc y el de salida, se muestra en la figura 3.4.

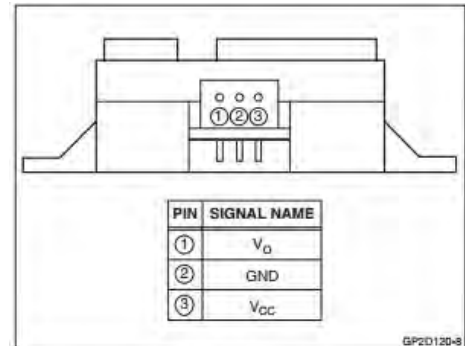


Figura 3.4: Conexión GP2D120

■ Esquema de conexiones del circuito

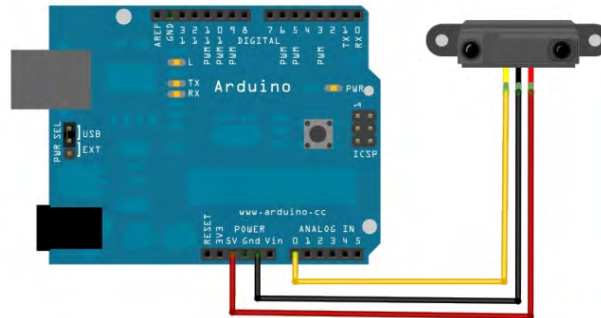


Figura 3.5: Esquema de conexión del Sistema de Detección de Proximidad

3.2.3. Sensor ADXL3xx

■ Diagrama de conexión

Para la conexión del ADXL3xx se ha de tener en cuenta el modelo que se tiene, en caso de ser 2g sólo se habrán de conectar a la placa dos ejes, normalmente el X e Y, sin embargo si es de 3g se deberán conectar cada uno de los ejes. El pin ST se refiere a Self Test, es decir, es para la calibración del sensor. En la figura 3.6 aparece como debe de ser la conexión.

- Esquema de conexión del circuito

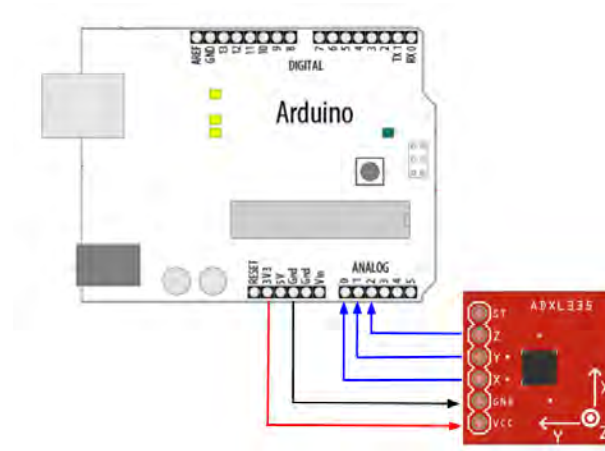


Figura 3.6: Esquema de conexión del ADXL3xx

3.2.4. Sensor ADXL345

- Diagrama de conexión

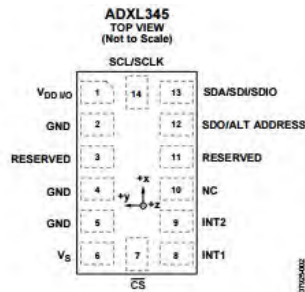


Figura 3.7: Pines del ADXL345.

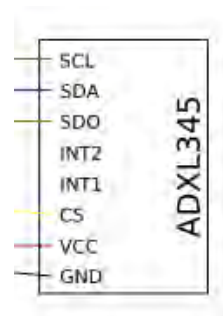


Figura 3.8: Conexiones ADXL345.

- Esquema de conexiones del ADXL345

Ya que existen varios posibles protocolos para el ADXL345, dependiendo de si se quiere conectar un único sensor o varios al tiempo, a continuación se mostrarán ambas conexiones.

Como se ve en la figura 3.9, las conexiones de este modelo es igual que el del modelo ADXL3xx, no necesita de ningún protocolo específico para su funcionamiento.

En caso de querer estudiar el funcionamiento de dos sensores ADXL345, deberán se configurados por el protocolo I2C y la librería *Wire*, que es la que lo implementa en el código.

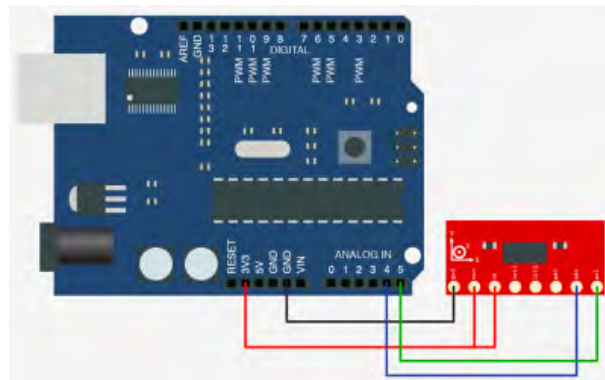


Figura 3.9: Esquema de un circuito simple para un ADXL345

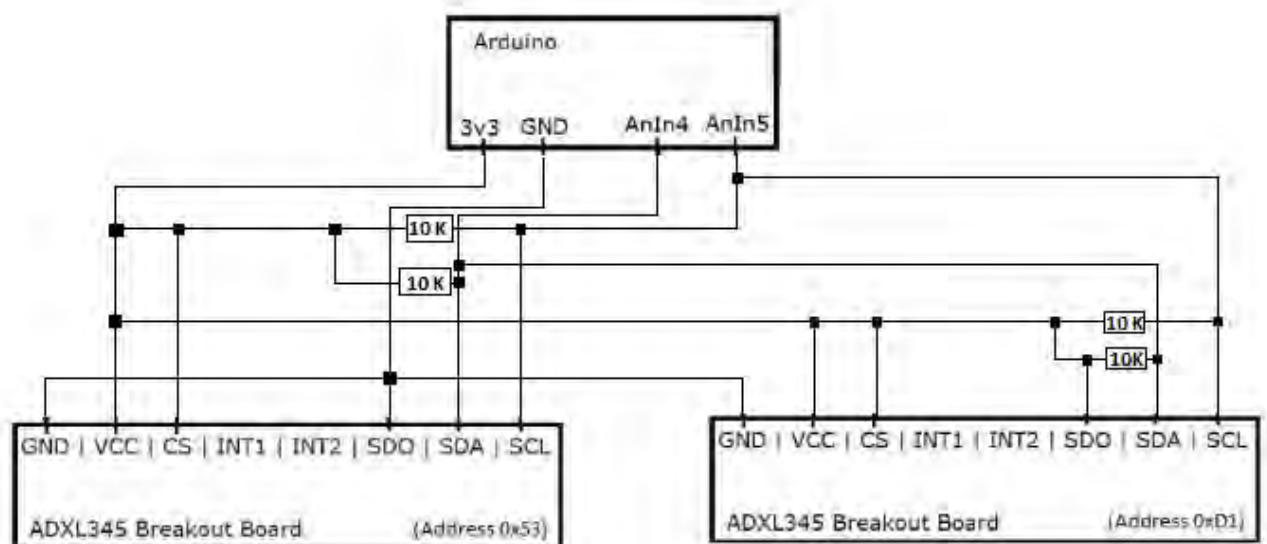


Figura 3.10: Esquema de un circuito doble para dos ADXL345

El protocolo I2C, tal como se muestra en la figura 3.10 tiene una propiedad llamada dirección del dispositivo, lo que permite la comunicación del dispositivo maestro a un dispositivo específico que estará conectado a los pines de datos y del reloj señalando el mismo la dirección. En la hoja de datos de ADXL345 explica que el sensor puede tener dos direcciones, dependiendo de como se cablea. Estas dos direcciones son 0x53 y 0xD1.

3.2.5. Sensores de contacto

■ Diagrama de conexión

Del esquema ilustrado en la figura 3.11, se deduce que el botón es simplemente dos cables paralelos y la conexión entre ambos sería el botón, según se apriete o no.

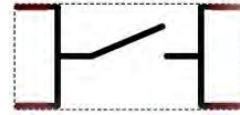


Figura 3.11: Conexión de un botón interruptor

■ Esquema de conexiones del circuito

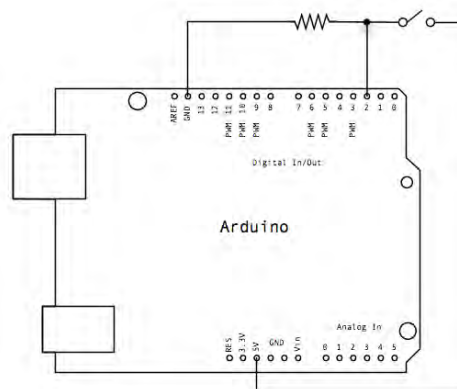


Figura 3.12: Esquema de conexión del circuito con un único botón.

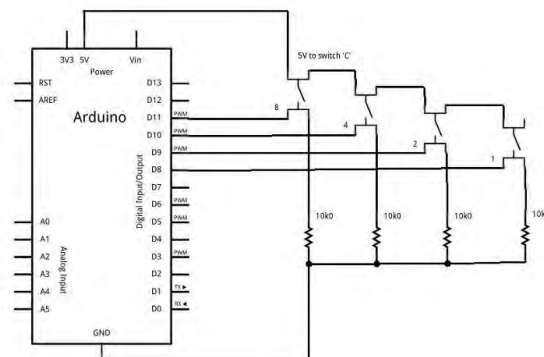


Figura 3.13: Esquema de conexión del circuito con 4 SWITCHES

3.2.6. Conexión MIDI a un Arduino

La conexión de un sintetizador y/o controlador MIDI se efectúa de la siguiente manera que aparece en la figura 3.14 Sin embargo, el sistema de detección ha de ser autónomo, para permitir

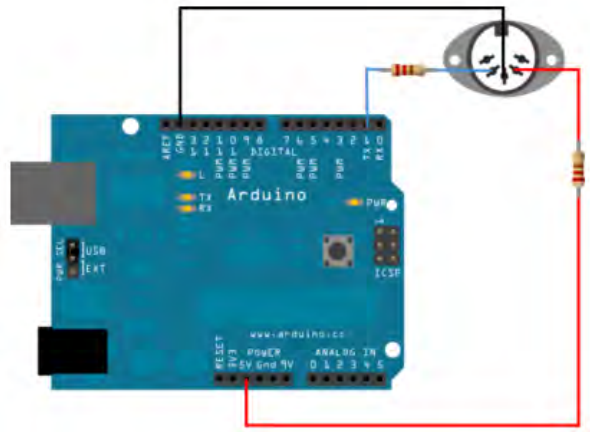


Figura 3.14: Conexión conector hembra MIDI a Arduino

una total movilidad al bailarín, es por ello que las conexiones se tienen que hacer a través del centro de control, un ordenador portátil, es por ellos que las conexiones que deberían seguir es la que aparece en la figura 3.15.

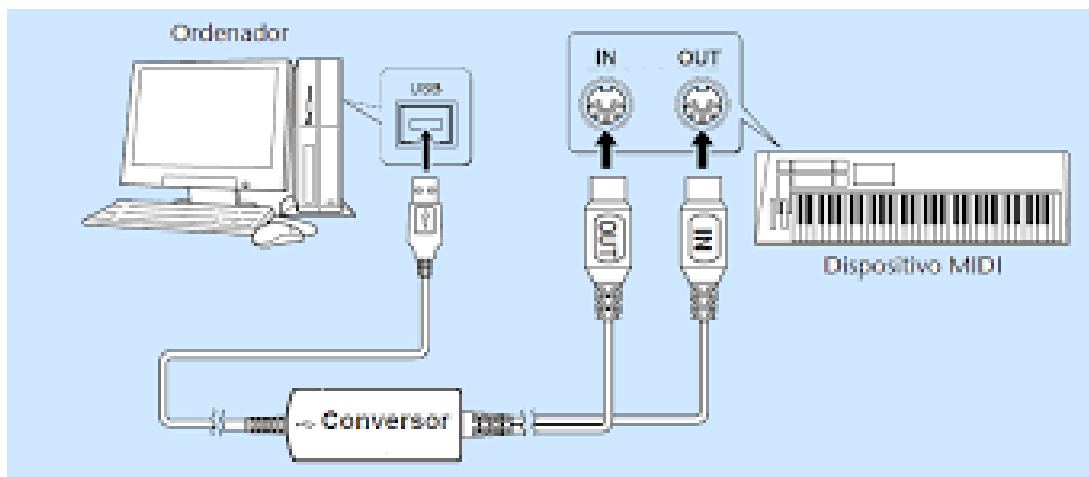


Figura 3.15: Conexiones entre ordenador y sintetizador

Capítulo 4

Desarrollo e implementación

4.1. Comunicación inalámbrica

Una vez que se han elegido los sensores que más se adecuan a nuestro sistema de detección, se ha de hacer un estudio de como se va a recibir la información ya que debe de ser inalámbrica, se sopesarán dos métodos, vía ZigBee o vía Bluetooth.

4.1.1. Introducción a XBee

Estos módulos, como aparecen en la figura 4.1 trabajan en la banda de 2.4GHz y usan el protocolo de comunicación IEEE 802.15.4. Habitualmente se utilizan en sistemas de seguridad, proyectos de automatización de hogares, monitorización de sistemas remotos, aparatos domésticos, alarmas contraincendios, plantas potabilizadoras de agua, etc. Son dispositivos que integran un transmisor- receptor de ZigBee junto con un procesador en el mismo.

■ Protocolo ZigBee

Fue creado por una organización sin ánimo de lucro, ZigBee Alliance, y desarrollada para crear un protocolo de red que se pudiera utilizar en una variedad de aplicaciones comerciales e industriales de baja velocidad de datos. Permite la comunicación entre componentes electrónicos de bajo consumo.

Opera en la banda de 2.4 Ghz y, a diferencia de bluetooth, no utiliza FHSS sino que emplea un único canal. El alcance varía según el tipo de antena usado y la potencia de emisión del dispositivo.



Figura 4.1: Módulo ZigBee (XBee)

4.1.2. Bluetooth

Es una tecnología inalámbrica de corto alcance, cuyo principal objetivo es la eliminación de cualquier tipo de cable de conexión.

Esta tecnología está diseñada para adaptarse a dispositivos de un tamaño pequeño y a un bajo coste. Su función es establecer redes de conexión mediante las cuales cualquier dispositivo puede intercambiar información con otro.

Sus características más destacables son su robustez, el bajo consumo de energía, la alimentación puede variar entre los 3 V a 9 V y bajo coste. También permite la interoperatividad y compatibilidad con cualquier otro dispositivo, con total independencia del fabricante.

El bluetooth opera en la banda libre de 2.4 GHz, o la llamada ISM (Industrial Scientific and Medical). Lo que permite a cualquier dispositivo bluetooth operar en cualquier parte del mundo.

■ Protocolo Bluetooth

La tecnología bluetooth viene definida un conjunto de protocolos, como se observa en el esquema de la figura 4.2 y una serie de perfiles. El concepto de perfil se refiere a una forma de comunicación o modelo de uso entre dos dispositivos bluetooth, siempre referido a la aplicación.

Los protocolos más comunes son algunos de los siguientes:

TCS: Telephony Control Specification

SDP: Service Discovery Protocol

L2CAP: Logical and Link Control Adaptation Protocol

HCI: Host Controller Interface

FHP: Freehand Profile

SDAP: Service Discovery Application Profile

SPP: Serial Port Profile

HSP: Head Set Profile

LANP: Local Area Network Profile

GOEP: General Object Exchange Profile

OBEX: Object Exchange

OPUSH: OBEX Object Push

FTRN: File Transfer

SYNC: Synchronization profile

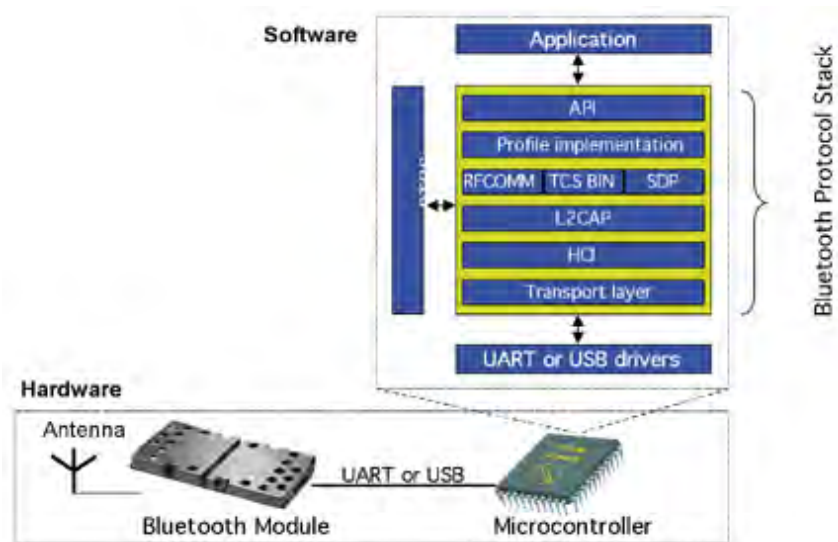


Figura 4.2: Esquema de protocolo Bluetooth

4.1.3. Comparación de tecnologías de comunicación

A continuación se incluye una tabla 4.1.3 donde se hace un breve resumen de ambas tecnologías de las que se ha tratado anteriormente.

Módelos	Bluetooth	ZigBee
Estándar	802.15.1	802.15.14
Frecuencia	2.4 GHz- 2.5 GHz	2.4 GHz- 2.5 GHz
Rango	10 - 100 m	100+ m
Consumo	Reducido (1/5 del consumo de WiFi)	Elevado
Ancho de banda	250 Kbps	720 Kbps
Usos	Usuario	Industrial
Seguridad	Obligatoria	Opcional

Tabla 4.1: Comparación ZigBee y EZ-Link.

Analizando las características de ambos sistemas de comunicación no se puede determinar que tecnología es mejor, sin embargo la elección adecuada dependerá de las especificaciones de diseño de la aplicación.

Los usos más comunes para ZigBee están más orientados al consumo industrial o para dispositivos totalmente autónomos sin apenas interacción con el usuario debido a su mayor rango de distancia y a la velocidad de transferencia de datos. Refiriendonos al bluetooth, este nos ofrece una tecnología flexible al permitir distintas topologías de red, son baratos y tiene un consumo relativamente bajo, lo suficiente como para alimentarse a través de baterías, es soportado por diferentes sistemas operativos y su fácil programación. También tiene un suficiente rango de proximidad.

En conclusión, viéndonos las especificaciones y usos que nos ofrece el bluetooth, es la tecnología más acertada para tener la autonomía que queremos para llevar a cabo el proyecto.

4.2. EZ - Link Adafruit

El módulo EZ-Link de Adafruit, es un bluetooth regular SPP de conexión serie que se puede conectar con cualquier ordenador o *tablet*. Aparece como un puerto serie más al conectarlo con el receptor deseado, cuya información se ha sacado de [8].

Una ventaja que tiene sobre otros módulos de bluetooth, es el hecho de detectar automáticamente y luego cambiar la velocidad de transmisión de serie todo dentro del módulo RF, es decir, es capaz de soportar las velocidades de transmisión más comunes: 2400, 4800, 9600, 19200, 38400, 57600, 115200 y 230400. También los controles de salida y entradas DTR/RTS/DSR están sincronizados automáticamente al puerto serie del ordenador. Esto nos permite trabajar con Arduino de forma inalámbrica y la conexión se realiza de forma automática, en cuanto el ordenador detecta el dispositivo simplemente se da a conectar en la zona de dispositivos de bluetooth y tras poner la contraseña, que al ser de fabrica es 1234, se conecta sin necesidad de programar para incluirlo en el código del sistema original.

El módulo bluetooth está compuesto por las siguientes partes, representadas en la imagen 4.3:

- **Módulo RF**

Se encarga de enviar toda la información del sistema en el que está conectado al ordenador y viceversa.

- **FTDI - Like Header**

Conexiones para el voltaje de entrada, y para los siguientes pines:

DSR: Se corresponde al llamado *Data Signal Ready*, y es pin de control de flujo por *hardware* de la señal de datos preparado.

Tx: Pin de salida del UART (*Universal Asynchronous Receiver/Transmitter*) serie, que se transmite desde el equipo asociado, de forma inalámbrica al módulo bluetooth.

Rx: transmite información del microcontrolador del EZ-link al equipo conectado.

DTR: Llamado *Data Terminal Ready*, es el pin de control de flujo de *hardware* que se envía desde el equipo emparejado al microcontrador del bluetooth.

- **Regulador**

Se encarga de convertir el voltaje de entrada, entre 3 a 16 VDC en 3 V, para la seguridad del dispositivo y usuario.

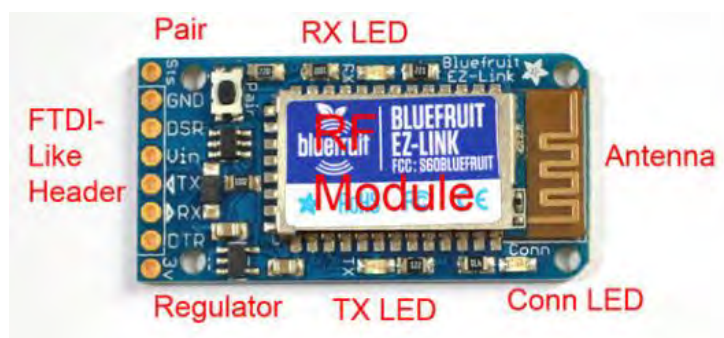


Figura 4.3: Componentes EZ- link Adafruit

Tal como se comentó en el apartado de los objetivos, este proyecto tiene como intención la medición de los movimientos de los brazos, piernas y torso. Cómo también tener algún sensor que permita al bailarín captar señales en la interacción con la escena.

Dependiendo del sensor que se utilice las señales obtenidas serán de distintos tipos. Por ejemplo, un telémetro acoplado en una mano permitirá, mediante los movimientos de dicha mano, capturar una señal de la distancia de la mano del bailarín a un objeto, sea una persona en escena, o un objeto inanimado. En el caso de usar giróscopos y/o acelerómetros, en función de la variación que tenga la mano, pie o torso respecto a un eje predefinido, provocará señales con distintas intensidades. Y así sucesivamente con cualquier sensor que exista.

4.3. Sistema de detección de proximidad

El sensor de proximidad es un transductor que detecta objetos que se encuentran a cierta distancia del sensor. Existen varios tipos según el principio físico en que se basan. Los más comunes son los interruptores de posición, los detectores capacitivos, los inductivos y los fotoelectricos, como el infrarrojos. También son fáciles de encontrar los sistemas telemétricos por láser y ultrasonido, SÓNAR.

El funcionamiento del **SÓNAR** es muy sencillo. Emite un tren de ultrasonidos a una frecuencia de 50kHz en un haz de forma cónica con una extensión de -15° a $+15^\circ$. Posee una duración del de 1 ms aproximadamente, y pasaría a modo receptor. Esto quiere decir que funciona como emisor y receptor con un único dispositivo.

Algunos de los problemas de este sensor es el ángulo de incidencia, puede provocar dobles rebotes o la amplitud del cono puede llevar a mediciones erróneas, los ruidos de otras fuentes de emisiónm atenuación por el medio depende de la frecuencia, temeperatura, etc.

El **Láser** trabaja con el mismo principio de tiempo de vuelo, devuelve un grupo de lecturas de 180° con una lectura por cada medio grado. Sin embargo hay que tener en cuenta el tipo de objeto que se quiere detectar y su distancia.

Por lo general los láseres usan el método de la triangulación para detectar la distancia con un objeto que esta relativamente cerca, mientras que con láseres de larga distancia se usa el método de comparación de fases. El Láser envía un haz de luz con diferentes frecuencias, y compara la señal que se ha reflejado en el objeto.

Los sensores **infrarrojos** se basan en un fototransistor o fotodiodo. La luz del IR (InfraRed) se transmite y luego se refleja del objeto en cuestión. El circuito de salida utiliza la señal del receptor para amplificarla y adaptarla a una salida que el sistema pueda entender. Se basan en el ángulo de reflexión, el sensor puede determinar la distancia hasta el objeto.

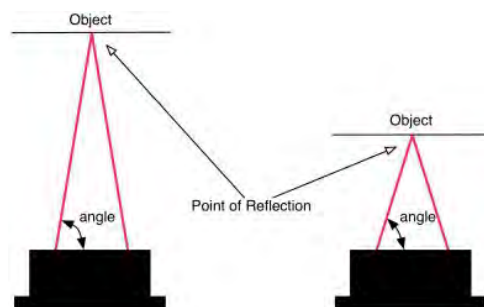


Figura 4.4: Ángulos de reflexión a distintas distancias

Como se puede observar en la figura 4.4, el ángulo de la luz reflejada está basada en la distancia del objeto. la parte receptora del IR es un objetivo de precisión que transmite la luz reflejada en varias partes de la matriz CCD⁹. Esta matriz puede entonces determinar el ángulo de incidencia de la luz reflejada. Este método permite que el sensor sea inmune a la interferencia de la luz ambiente y permite cualquier color del objeto que está siendo detectado.

⁹ *textCharge Coupled Device* - Se encarga de la conversión de una señal luminosa en una señal eléctrica

Este proyecto se centra en la proximidad de las manos del bailarín, teniendo en cuenta las posibles opciones de sensores, se optó por el infrarrojos ya que es pequeño, tiene poco consumo de energía y tiene variedad en los opciones de salida.

4.3.1. Sensor de Infrarrojos SHARP IR GP2D120

■ Descripción

Existen varios modelos de IR SHARP, sin embargo se eligió el de menor tamaño físico. Esto también provoca que el rango de distancia máximo sea menor que el resto de modelos.



Figura 4.5: IR SHARP GP2D120

En este caso, el SHARP GP2D120, que aparece en la imagen 4.5 tiene una procesador de señal integrado y una salida analógica cuyo rango mínimo es de 4 cm y como máximo de 30 cm. Debido a la trigonometría que se usa para el cálculo de la distancia a un objeto basado en el ángulo de incidencia de luz reflejada, la salida de estos detectores no es lineal con respecto a la distancia. A continuación se muestra en un gráfico, figura 4.6 la salida en función a la distancia que se está midiendo de un objeto.

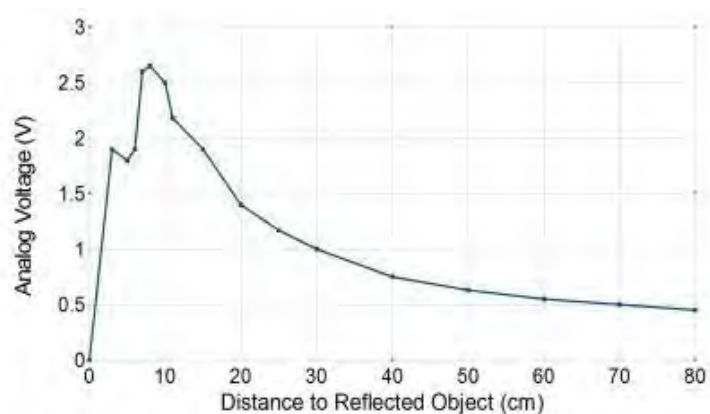


Figura 4.6: Ejemplo de gráfica para sensor IR SHARP

■ Características

- Bajo coste
- Fácil calibración para obtener resultados lineales
- Rango entre 4 - 10 cm
- Tiempo de respuesta de 39 ms
- Tiempo de retraso de 44 ms
- Consumo medio de 33 a 50 mA
- Voltaje maximo de 4,5 a 5,5 V
- Temperatura máxima de trabajo de -10 a 60 °C

■ Código Arduino

```
#include <SharpIR.h>

const int Pin = A1;
int val = 0;

void setup() {
  pinMode(Pin, INPUT);
}

void loop() {
  val = analogRead (Pin);
  int Distance = 2076/(val - 11);
  Serial.print(Distance);
  Serial.println("cm");
  delay(1000);
}
```

Al principio del código se le incluye la librería *SharpIR.h*, que nos facilita el leer datos de forma continua y constante. Nos ayuda a crear un valor con la media entre x valores que se han tomado con el IR.

Se inicializa el Pin analógico al que vamos a conectar, dándole un valor inicial de 0 para evitar que haya ningún otro valor distinto a este. Tiene que ser configurado como un Pin de entrada nada comenzar el traje a funcionar. Para finalizar, dado que los valores que obtenemos con la lectura tienen unidades en voltios [V], se han de convertir esos valores a cm con la siguiente fórmula:

$$Distance = \frac{2076}{val - 11} \quad (4.1)$$

4.4. Sistema de detección de posición

En este apartado, para la detección de posición se hizo uso de acelerómetros. Se eligieron por ser baratas, fáciles y fiables para pequeños proyectos electrónicos.

Un acelerómetro, como su nombre indica, mide la aceleración. Está formado por diminutas micro-estructuras que se doblan debido a la inercia y la gravedad. Al sufrir una deformación por alguna de las dos causas anteriores, se mide cuanto se ha doblado por una señal eléctrica. Las micro-estructuras sólo pueden medir la fuerza en un eje de aceleración, por lo que para tener una medida más precisa se ha de hacer uso de acelerómetros que midan de al menos 2 ejes. Refiriendonos a la gravedad, como cualquier otro objeto en la tierra, este está sujeto a una aceleración de $-9,81m/s^2$. Gracias a esto el sensor puede detectar qué ángulo tiene con respecto a la gravedad, que lo mide según el gráfico que aparece en la figura 4.7.

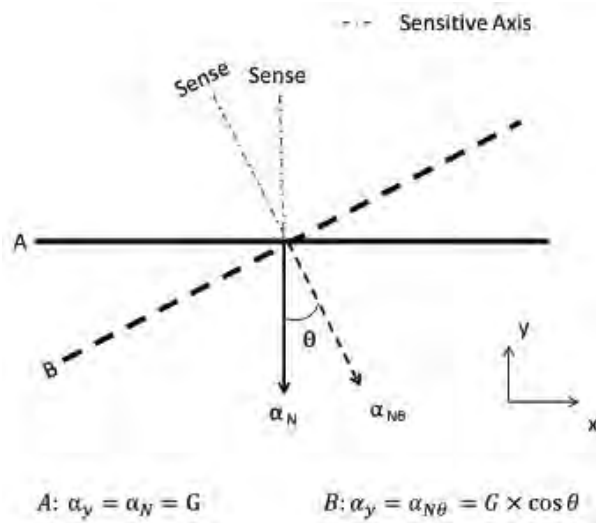


Figura 4.7: Medida de la gravedad del Acelerómetro

Cuando se hace uso de un acelerómetro, en las características se incluye la calificación de G. Quiere decir la fuerza máxima de gravedad que el sensor puede dar como señal. A menor número de g, más sensible será a los cambios en el movimiento, pero esto hace que también sea más sensible a la interferencia de cualquier vibración. Para calcular la magnitud de la aceleración de un acelerómetro de dos y tres ejes:

$$Acceleration = \sqrt{x^2 + y^2} \text{ Para un acelerómetro de 2g} \quad (4.2)$$

$$Acceleration = \sqrt{x^2 + y^2 + z^2} \text{ Para un acelerómetro de 3g} \quad (4.3)$$

Para el cálculo de la fuerza detectada en un acelerómetro, se utiliza la siguiente ecuación:

$$Force = -g * \cos x \quad (4.4)$$

La función de la inclinación del ángulo medida por los acelerómetros no es lineal para ángulos muy grandes, esto se debe al salto de la sensibilidad cuando los valores medidos de la gravedad se vuelven muy pequeños. El comportamiento de la función se vuelve asintótica con ángulos grandes, como se muestra en la figura 4.8.

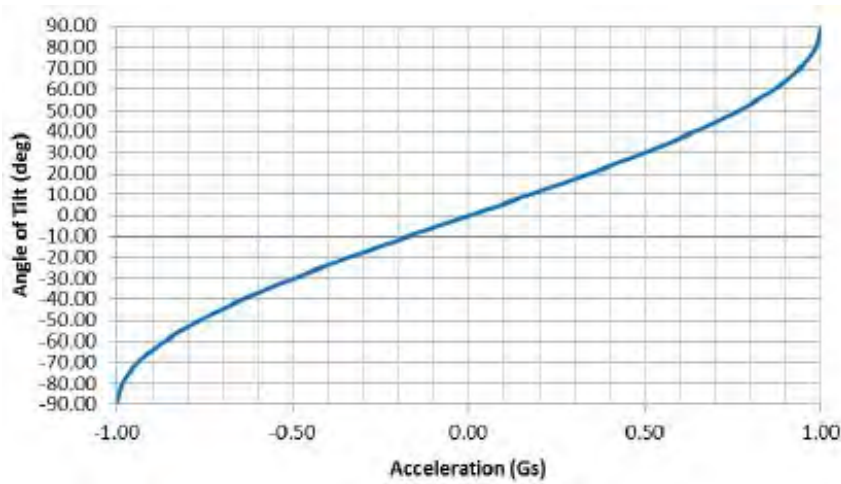


Figura 4.8: Gráfica de ángulo de inclinación frente a la aceleración de la gravedad

La gráfica del ángulo de aceleración se relaciona con el ratio de la aceleración debido a la gravedad, de aquí obtenemos la siguiente relación:

$$Angulo = \arcsin \frac{Aceleracion[Gs]}{1[G]} x \frac{180}{3,14} \quad (4.5)$$

4.4.1. Acelerómetro ADXL 3xx

■ Descripción

Este acelerómetro engloba distintos modelos de Sparkfun accelerometers, el ADXL320, ADXL321, ADXL322 y el ADXL330. Todos estos acelerómetros tienen salida analógica entre 0 y 5 voltios.

Son dispositivos pequeños, finos (tal como se observa en la imagen 4.17), con poco consumo energético c on 3 ejes. El sensor mide la aceleración en un rango de -3 a +3 g. La señal de salida es analógica y proporcional a la aceleración. Puede medir la aceleración estática de la gravedad mediante la detección de inclinación, cmo la aceleración dinámica que resulta del movimiento, golpe o vibración.



Figura 4.9: Acelerómetro ADXL3xx

La placa de Arduino mide el voltaje de salida del acelerómetro en modo analógico, por lo que la señal leía desde el pin analógico debe convertir ese dato en digital, utilizando el conversor analógico - digital de 10 bits. Los pines analógicos leen valores enteros de 0 a 5 V que se corresponden a valores entre 0 y 1023. El siguiente procedimiento se ha desarrollado para convertir la señal analógica a unidades de aceleración. La sensibilidad de este acelerómetro es de 420 mV/g nominal con un voltaje de entrada de 3.3 V, que indica que 1g de aceleración es igual a 420 mV/g. Podemos obtener la resolución del acelerómetro con la siguiente fórmula:

$$\frac{5V}{1024} = 0,00488V = 4,88mV \quad (4.6)$$

El ADXL3xx utiliza una estructura simple para sentir los ejes X, Y y Z. Como resultado, los tres ejes están situados ortogonalmente, como se ve en la figura 4.10 y tienen poca sensibilidad en el cruce de los ejes. Para obtener resultados óptimos, el sensor se tiene que calibrar previamente fuera del sistema.

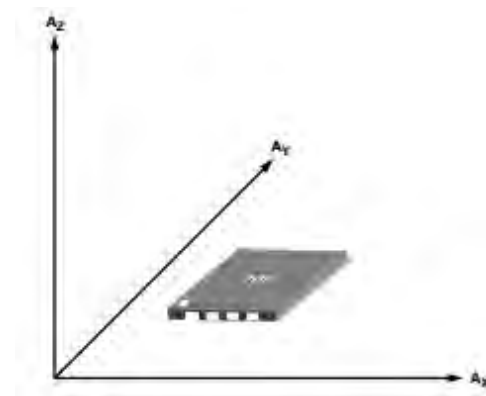


Figura 4.10: Ejes de aceleración en relación con la sensibilidad.

Como se ha explicado anteriormente, el acelerómetro tiene un comportamiento no lineal. Sin embargo el ADXL 3xx es más sensible a la inclinación cuando el plano del circuito está perpendicularmente a la gravedad. El grado de sensibilidad se tiene en cuenta al usar la gráfica del ángulo de inclinación. En la función lineal, figura 4.11, se muestra a partir de los dos valores que nos da la hoja de datos del ADXL322 (17.5 mg/° para menor que 45° y 12.2 mg/° para mayor que 45°).

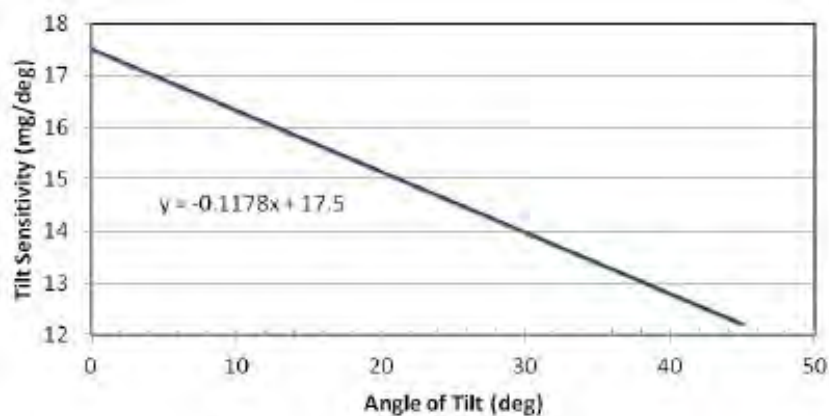


Figura 4.11: Gráfica del ángulo de inclinación frente a la función de sensibilidad

■ Características

- Voltaje entre 1,8 y 3,6 V
- Corriente 320 uA
- Salida analógica

- Rango de detección 3 ejes, -3 a +3 g
- Máximo rango de fuerza 10.000 g
- El ancho de banda se configura a través de condensadores externos en cada uno de los ejes

■ Código Arduino

Como en el sensor anterior, se ha de especificar que pines de entrada serán usados por cada eje para la lectura de datos. Estos pines se les declara como entrada cuyo valor será analógico, por lo que se tiene que declarar en la lectura de datos que tipo de lectura será, por ello se utiliza la función *analogRead()*.

```
const int xpin = A3;
const int ypin = A2;

void setup() {
  pinMode(xpin, INPUT);
  pinMode(ypin, INPUT);
}

void loop() {
  Serial.print(analogRead(xpin));
  Serial.print("_");
  delay(500);

  Serial.print(analogRead(ypin));
  Serial.print("/");
  delay(1000);
}
```

4.4.2. Acelerómetro ADXL 345

■ Descripción

Las características tanto físicas como específicas son muy similares al ADXL3xx. El acelerómetro ADXL345 es de pequeñas dimensiones, así se puede ver en la imagen 4.18, con un bajo consumo de energía, consta de 3 ejes de alta resolución y tiene hasta -16 a + 16g, como ve en la respuesta de salida frente a la g, que aparece en la figura 4.13. El configurable según dos protocolos, a través del SPI (3 cables o 4 cables) o a través de la interfase digital I2C. En este proyecto se utilizó la segunda opción.

Como otros acelerómetros, puede medir la aceleración estática de la gravedad en aplicaciones de mediciones de inclinación, al igual que aceleraciones dinámicas cuyo resultado viene dado por movimientos o choques. Su alta resolución permite la detección de los cambios en la inclinación de objetos de menos de 1°.



Figura 4.12: Acelerómetro ADXL345

Además de lo básico incluye otras muchas funciones de mediciones especiales. Como la medición de inactividad, es decir la falta de movimiento, o si la aceleración en cualquiera de sus ejes excede el nivel configurado por el usuario. También detecta si el objeto está cayendo. Todo esto están configurados para un bajo consumo y gracias a los estudios de las mediciones de umbrales y aceleración activa evita la disipación de energía.

■ Características

- Suministro de voltaje de 2 a 3,6 VDC
- Corriente de entrada muy baja 0,1 uA
- Resolución muy alta
- Monitorización de actividad/inactividad
- Detección de pequeño golpes simples o dobles
- Detección de caída libre

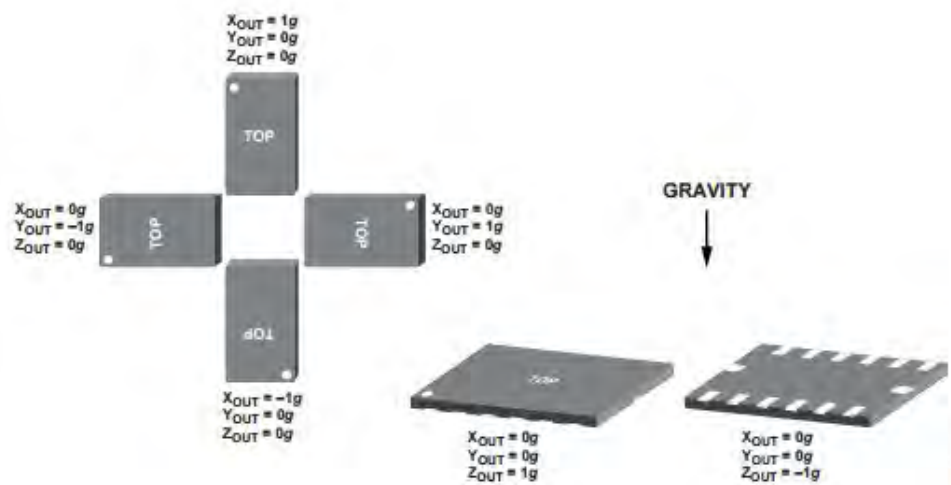


Figura 4.13: Respuesta de salida frente a la orientación de la g.

- Interfaces SPI e I2C
- Temperatura desde -40 a 85 °C

■ Código Arduino

```
#include <Wire.h>

#define DEVICE_A (0x1D)
#define DEVICE_B (0x53)
#define TO_READ (6)

byte buff[TO_READ] ;
char str[512];

void setup() {
  Wire.begin();
  writeTo(DEVICE_A, 0x2D, 24);
  writeTo(DEVICE_B, 0x2D, 24);
}

void loop() {
  readFrom(DEVICE_A, regAddress, TO_READ, buff);
  xa = (((int) buff[1]) << 8) | buff[0];
  ya = (((int) buff[3]) << 8) | buff[2];
```

```

    za = (((int) buff[5]) << 8) | buff[4];

    readFrom(DEVICE_B, regAddress, TO_READ, buff);
    xb = (((int) buff[1]) << 8) | buff[0];
    yb = (((int) buff[3]) << 8) | buff[2];
    zb = (((int) buff[5]) << 8) | buff[4];

    sprintf(str, "%d_%d_%d_%d_%d_%d", xa, ya, za, xb, yb, zb);
    Serial.print(str);
    Serial.write(10);
    Serial.print("/");
    delay(1000);
}

void writeTo(int device, byte address, byte val) {
    Wire.beginTransmission(device);
    Wire.write(address);
    Wire.write(val);
    Wire.endTransmission();
}

void readFrom(int device, byte address, int num, byte buff[]) {
    Wire.beginTransmission(device);
    Wire.write(address);
    Wire.endTransmission();
    Wire.beginTransmission(device);
    Wire.requestFrom(device, num);

    int i = 0;
    while(Wire.available())
    {
        buff[i] = Wire.read();
        i++;
    }
    Wire.endTransmission();
}

```

El funcionamiento de este código se basa en el protocolo I2C, y para la inicialización se debe incluir una librería específica llamada *Wire.h*. Además de esto, también se tiene que

declarar cual de los dos acelerómetros sera el principal y cual el secundario, definiendo sus direcciones correspondientes, y el número total de bytes que se leerán por cada eje, en nuestro caso son 6, ya que tendremos 2 valores por cada eje.

Nada más comenzar el programa, se inicializa la librería *Wire.h*, donde crea la conexión entre las dos direcciones para el protocolo I2C, y se encienden ambos ADXL345 con las funciones *writeTo()*. Esta función inicializa la transmisión de datos a la p placa, le envía la dirección registrada, el valor para escribir y finalmente termina la transmisión.

En el main code, se lee la información obtenida por ambos acelerómetros con la función *readFrom()*. Como en el caso de la función de escribir, esta función comienza la transmisión al aparato, envía la dirección a leer desde, y terminaría la transmisión. A parte tiene que pedir al dispositivo los 6 bytes, para ello está la función *Wire.requestFrom()*, a partir de aquí introducimos una función de loop, *while()* para recibir tantos bytes como hayas configurado en un principio. Cada eje tiene una resolución de 10 bits, como por ejemplo 2 bytes, por ello tenemos que convertir ambos bytes en números enteros:

```
xa = (((int) buff[1]) << 8) | buff[0];  
ya = (((int) buff[3]) << 8) | buff[2];  
za = (((int) buff[5]) << 8) | buff[4];
```

4.5. Sistema de detección de contacto

Son sensores utilizados para obtener información asociada con el contacto entre la mano manipuladora y objetos en el espacio de trabajo. Los objetivos principales son proporcionar información sobre los objetos en el entorno del bailarín, especialmente la presencia de algún objeto en contacto con el sensor de contacto o bien la fuerza que ejerce la mano del bailarín al sensor.

Para considerar un sensor de contacto, este se debe interrumpir un contacto eléctrico por medio de una fuerza externa. Existen muchos modelos de sensores de contacto, dependiendo de la salida que se quiera obtener, los más conocidos y usados son los pulsadores, interruptores e interruptores con corrección de rebote. En nuestro caso se construyó desde cero un botón pulsador, ya que queremos una salida binaria, por lo que usaremos los pines digitales de la placa Arduino.

Un sensor de contacto es un dispositivo simple que completa un circuito. Un extremo del botón está conectado a la fuente, por lo general de 5 V es decir BT (Baja Tensión), y el otro a un pin digital. Cuando se prensa el sensor el circuito se abre o cierra, según esté configurado en el código del programa. El pin digital simplemente devuelve si hay 5 V o 0 V.

Las ventajas que se pueden encontrar en este tipo de sensores son algunas de las siguientes:

- Pueden detectar cualquier objeto independientemente del material con el que estén realizados
- No les afectan interferencias procedentes del medio exterior
- Su salida está constituida por uno o más contactos libres de potencial
- Funcionamiento exclusivamente mecánico

Sin embargo hay un problema latente con los botones pulsadores, y es el hecho de tener efecto rebote en el encendido o apagado, como aparece en la figura 4.14, al igual que es muy sensible a las vibraciones externas, se ha de tener en cuenta para minimizar el error al máximo posible.

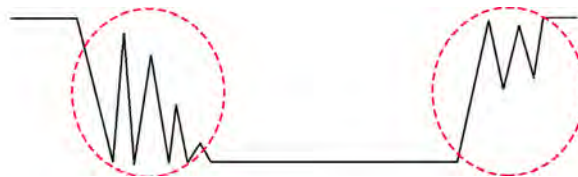


Figura 4.14: Gráfico del efecto rebote en el ON y OFF

Sensor de contacto BOTÓN INTERRUPTOR

■ Descripción

Son un tipo de elementos externos que nos permite controlar físicamente acciones a realizar por nuestro sistema a través de dar continuidad o no a la señal de entrada en la que están instalados. Son muy sencillos de usar y programar.

El sistema de contacto consta de 4 interruptores colocados en el pecho y espalda del bailarín. Cada sensor cuenta con dos placas hechas de planchas de un espesor de 0.06mm y de latón. Se usaron placas de latón debido a sus características e

■ Características

A partir de la siguiente tabla 4.5, se optó por la utilización del latón, por su bajo coste, tiene una buena conductividad, su densidad nos permite amoldar las placas a la forma que se desee, y es posible soldar los cables de cobre en la misma placa sin que se llegue a fundir el latón.

Metales	Densidad	Conductividad	Temperatura de fusión
	$[kg/dm^3]$	$[Sm/mm^2]$	$[C]$
Cobre	8,7 - 8,9	58,5	1.083
Aluminio	2,69 - 2,7	35,85	657 - 660
Latón	8,4 - 8,7	25	900 - 960
Zinc	7,1	16.6	419,4 - 430
Niquel	8,8 - 8,9	11,6	1455
Estaño	7.3	9.17	232

Tabla 4.2: Principales características de algunos metales

■ Código Arduino

```
void setup() {
  pinMode(4, INPUT_PULLUP);
  pinMode(5, INPUT_PULLUP);
  pinMode(6, INPUT_PULLUP);
  pinMode(7, INPUT_PULLUP);
  pinMode(13, OUTPUT);
}

void loop() {
  int S1 = digitalRead(4);
  int S2 = digitalRead(4);
  int S3 = digitalRead(4);
  int S4 = digitalRead(4);

  Serial.print(S1);
  Serial.print(S2);
  Serial.print(S3);
  Serial.print(S4);
  Serial.print("/");

  if (S1 == HIGH) {
    digitalWrite(13, LOW);
  }
  else {
    digitalWrite(13, HIGH);
  }
  if (S2 == HIGH) {
    digitalWrite(13, LOW);
  }
  else {
    digitalWrite(13, HIGH);
  }
  if (S3 == HIGH) {
    digitalWrite(13, LOW);
  }
  else {
    digitalWrite(13, HIGH);
  }
}
```

```
    if (S4 == HIGH) {  
        digitalWrite(13, LOW);  
    }  
    else {  
        digitalWrite(13, HIGH);  
    }  
  
    delay (1000)  
}
```

Básicamente en la inicialización del programa se declaran que pines son y de que tipo, en este caso se han configurado como 'Pull-Up'.

El estado de Pull-Up implica que la lógica de la función está invertida. Cuando esta abierto el circuito esta en HIGH y cuando se cierra se vuelve al estado LOW.

En cuanto a al main code, con la función *digitalRead()* lee los datos y los guarda en una variable. Para verificar en que estado se encuentra cada 'Switch' o botón, se usa una función condicional, *if()* y *else()*, así se encenderá el LED que se encuentra en el pin 13 de la placa de Arduino.

4.6. Integración sensorial en un sistema de control

Una vez que se han explicado cada uno de los sensores y el tipo de conexión que tendrán con el ordenador, queda hacer el conjunto de todo el sistema, para poder después ver los resultados obtenidos y las conclusiones que se han sacado de todo el proyecto.

4.6.1. Equipamiento

Para la puesta en marcha de nuestro sistema se han necesitado los siguientes, materiales, componentes y dispositivos que se listan a continuación, utilizados para la alimentación, conexión de componentes, calibración, montaje, comunicación inalámbrica y experimentos del sistema.

- Fuente de alimentación de 9 V, proporciona energía a todo el sistema, placa y sensores, figura 4.15.



Figura 4.15: Pila de 9 V

- Una placa Arduino UNO más cable adaptador, figura 4.16.



Figura 4.16: Arduino UNO + cable

- Acelerómetro ADXL 3xx, figura 4.17
- Dos Acelerómetros ADXL 345, figura 4.18



Figura 4.17: ADXL 3xx



Figura 4.18: ADXL 345

- Módulo bluetooth EZ - link Adafruit, incluye condensador de 1uF, conector para batería, pines macho y hembra, figura 4.19

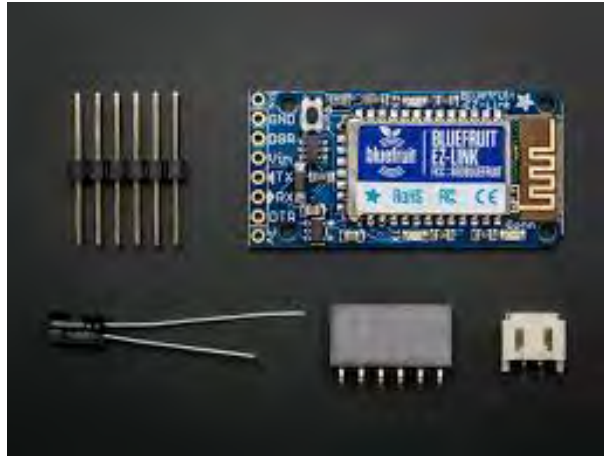


Figura 4.19: EZ - Link módulo

- Sensor IR SHARP GP2D120, figura 4.20



Figura 4.20: Sensor IR SHARP

- Planchas de Latón con un espesor de 0.06 mm, figura 4.21

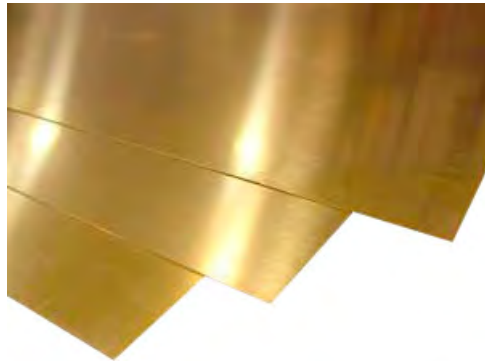


Figura 4.21: Láminas de Latón

- Neopreno para crear los Switches, y el bolsillo donde va situada la placa con todos los conectores, figura 4.24



Figura 4.24: Tela de neopreno

- Protoboard, figura 4.22



Figura 4.22: Protoboard

- Baquelita universal breadboard, figura 4.23



Figura 4.23: Baquelita para soldar

- Muñequera y tobillera con sensores de proximidad y posición, figura 4.25
- Faja donde se encuentra la placa y el ADXL 3xx, figura 4.26



Figura 4.25: Muñequera y tobillera



Figura 4.26: Faja de neopreno

- Cables de distintos grosores, tamaños y colores para las conexiones, figura 4.27



Figura 4.27: Cables

- Aguja e hilo para coser todos los sensores al traje, figura 4.28
- Soldador con bobina de estaño para soldar todas las conexiones posibles, figura 4.29



Figura 4.28: Aguja e hilo negro



Figura 4.29: Soldador y estaño

- Multímetro para comprobar que el sistema estaba bien conectado y soldado, figura 4.30



Figura 4.30: Multímetro

- Ordenador Portátil para la recepción de datos, registro y representación física, figura 4.31



Figura 4.31: Ordenador portátil hp

- Software instalado en el PC:
 - Arduino IDE, para la programación de Arduino UNO
 - CoolTerm, para la lectura de los datos oobtenidos por el sistema
 - LaTeX, para la escritura del informe
 - OrCAD Capture CIS, para el diseño de las PCBs y el circuito
 - Frizting, para sacar el esquema del sistema de detección

4.6.2. Descripción del funcionamiento

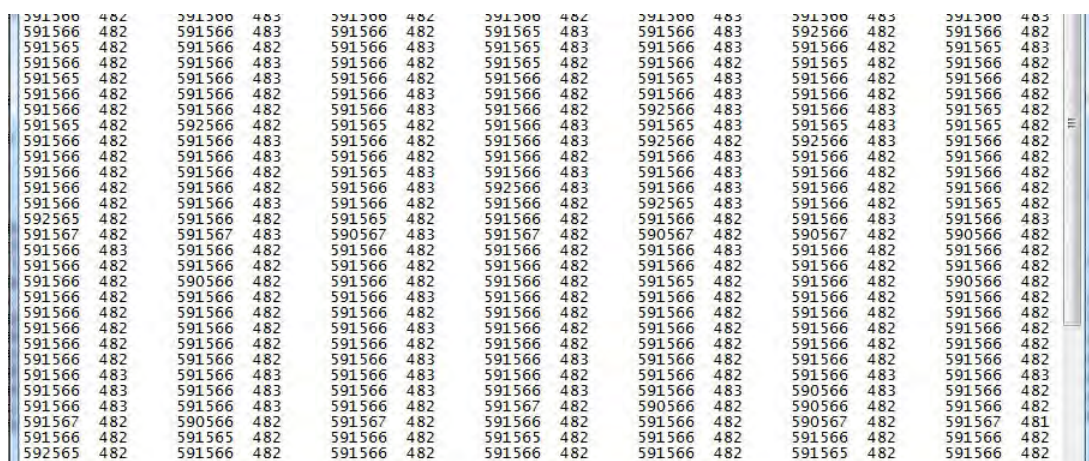
El sistema que se ha desarrollado tiene como objetivo el control del espacio que tiene el bailarín a su alrededor, es decir, luces, sonidos, etc.

En principio el sistema está listo para usarse, ya que una vez calibrados los acelerómetros se pueden usar tantas veces como se desee que el valor de los datos obtenidos no va a variar. Aunque, en los ensayos que se están llevando a cabo, si que se están haciendo mediciones y recalibrando los sensores para llegar a los valores que queremos.

Entre el bluetooth que está conectado la placa, y el ordenador con el que se quiere establecer una conexión, se establece una comunicación inalámbrica que permitirá visualizar los valores muestreados durante la detección de los parámetros.

Con un muestreo de 1 segundo de periodo, los sensores del sistema analizan la posición, la proximidad y en caso de haber algún contacto con el traje los datos cuantificados por Arduino, dónde a través de señales MIDI se codificarán para que tengan la respuesta que se haya configurado previamente. Esta representación, permite observar los cambios que sufre el bailarín en el entorno en el que se está desarrollando, de manera casi instantánea.

Además de la visualización insitu de los datos generados, también se genera de manera automática un fichero en formato .txt, la salida sería como la que aparece en la imagen 4.32, donde se recopilan todos los datos obtenidos. De modo que podría hacerse una evaluación posterior más exhaustiva si se considerase útil o necesaria.



391300	482	391300	483	391300	482	391300	482	391300	483	391300	483	391300	483
591566	482	591566	483	591566	482	591565	483	591566	483	592566	482	591566	482
591565	482	591566	482	591566	483	591565	483	591566	483	591566	482	591565	483
591566	482	591566	483	591566	482	591565	482	591566	482	591565	482	591566	482
591565	482	591566	483	591566	482	591566	482	591565	483	591566	482	591566	482
591566	482	591566	482	591566	483	591566	482	591566	483	591566	482	591566	482
591566	482	591566	482	591566	483	591566	482	592566	483	591566	483	591565	482
591565	482	592566	482	591565	482	591566	483	591565	483	591565	483	591565	482
591566	482	591566	483	591566	482	591566	483	592566	482	592566	483	591566	482
591566	482	591566	483	591566	482	591566	482	591566	483	591566	482	591566	482
591566	482	591566	482	591565	483	591566	483	591566	483	591566	482	591566	482
591566	482	591566	482	591566	483	592566	483	591566	483	591566	482	591566	482
591566	482	591566	483	591566	482	591566	482	592565	483	591566	482	591565	482
592565	482	591566	482	591565	482	591566	482	591566	482	591566	483	591566	483
591567	482	591567	483	590567	483	591567	482	590567	482	590567	482	590566	482
591566	483	591566	482	591566	482	591566	482	591566	483	591566	482	591566	482
591566	482	591566	482	591566	482	591566	482	591566	482	591566	482	591566	482
591566	482	590566	482	591566	482	591566	482	591565	482	591566	482	590566	482
591566	482	591566	482	591566	483	591566	482	591566	482	591566	482	591566	482
591566	482	591566	482	591566	482	591566	482	591566	482	591566	482	591566	482
591566	482	591566	482	591566	483	591566	482	591566	482	591566	482	591566	482
591566	483	591566	483	591566	483	591566	482	591566	482	591566	483	591566	483
591566	483	591566	483	591566	483	591566	483	591566	483	590566	483	591566	482
591566	483	591566	483	591566	482	591567	482	590566	482	590566	482	591566	482
591567	482	590566	482	591567	482	591566	482	591566	482	590567	482	591567	481
591566	482	591565	482	591566	482	591565	482	591566	482	591566	482	591566	482
592565	482	591566	482	591566	482	591566	482	591566	482	591565	482	591566	482

Figura 4.32: Resitro de muestra en CoolTerm de un giroscopo

Cabe mencionar que, mientras que la detección de datos es un proceso continuo e ininterrumpido, a menos que se apague el sistema, desde la inicialización del sistema, la obtención de datos y el registro de resultados, este último puede pararse e iniciarse de nuevo sin que llegue a afectar a la detección de datos, que seguirá operando, a no ser que sea desconectado, como se ha comentado anteriormente.

4.6.3. Montaje

El sistema de sensores final se compone de los cuatro sistemas individuales descritos en la memoria (4.4, 4.5 y 4.6), que operan de forma independiente, de modo que tanto los errores en el funcionamiento como en el conexionado pueda controlarse y solventarse de manera también independiente. También con el objetivo de minimizar errores de conexión y para facilitar el desentrañar las conexiones, se ha mantenido un código de colores en el cableado (en la medida de lo posible), como la misma posición de los conectores a la hora de conectarlos a la PCB. Todo esto se ha llevado a cabo desde el diseño inicial en OrCAD Capture CIS hasta el montaje final del hardware, para facilitar el seguimiento y la revisión.

- Cable rojo: alimentación acelerómetro o IR, ya sea 5 V o 3,3 V
- Cable negro: GND - Tierra (0 V)
- Cable blanco: datos de entrada del ADXL 345
- Cable verde : datos de entrada del ADXL 345
- Color verde claro: datos de entrada del ADXL 3xx
- Cable amarillo claro: datos de entrada del ADXL 3xx
- Cable amarillo: datos de entrada del IR
- Cable marrón: datos pines digitales
- Cable naranja: alimentación Switches

Como se ha mostrado en la sección de cada uno de los subsistemas, cada sensor se tiene que conectar a un pin analógico o digital de la placa Arduino para posteriormente enviarlos al ordenador y procesar esos datos.

- Proximidad: pin analógico A2.

- Posición:

ADXL 3xx: pines analógicos A0 y A1, para las coordenadas X e Y respectivamente.

ADXL 345: pines analógicos A4 y A5, corresponden a las coordenadas X e Y respectivamente.

- Switches: pines digitales desde D7 al D3.

En la figura 4.33 se muestra el esquema final de conexiones del prototipo de nuestro sistema de detección.

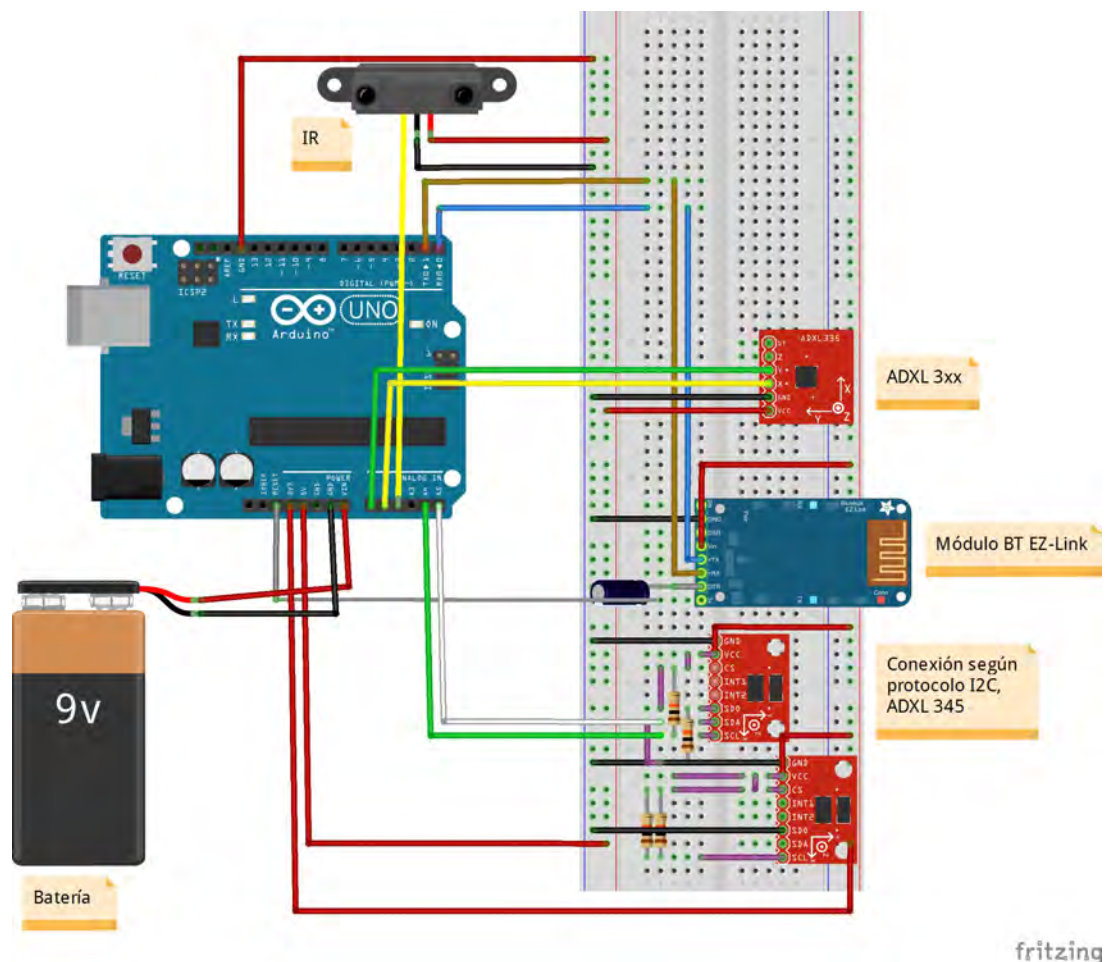


Figura 4.33: Esquema de conexiones de todo el sistema

En la figura 4.34 se muestra como quedó la placa junto con el bluetooth y cada uno de los conectores de los sensores:

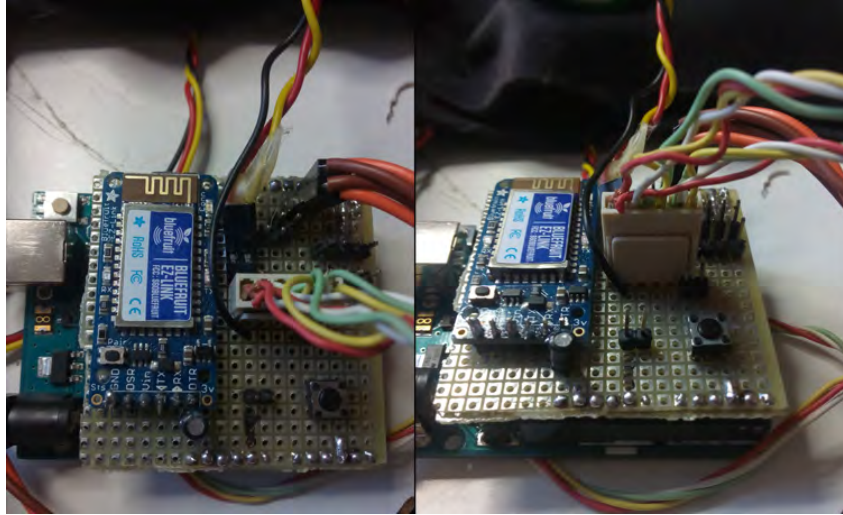


Figura 4.34: Placa Arduino con módulo externo para conectores

Resultados y experimentos

En este capítulo se realiza una descripción de los principales resultados obtenidos del funcionamiento del traje. El traje, como se ha explicado en la introducción es parte de un trabajo en proceso de desarrollo musical a partir del movimiento. Se enmarca, por tanto, dentro de un proyecto de investigación de gran envergadura realizado por el artista Javier Aparicio.

Este proyecto no tiene como objetivo final el obtener un resultado exacto, si no todo lo contrario, el poder expresar a través de un movimiento una variedad infinita de emociones y sentimientos transmitidas a través de sonidos e imágenes. Sin embargo es importante recalcar que durante el proceso de desarrollo se ha ido generando pruebas para comprobar que el funcionamiento teórico es el esperado en la realidad. El testeo se ha hecho para determinar la exactitud de todos los datos obtenidos para cualquier posición que pueda llegar a tener el sistema. Una vez verificado que toda la información leída por cada uno de los sensores individualmente y en conjunto era la adecuada, se dió por finalizado el muestreo y testeo de datos.

Experimentalmente, cuando se probó el traje en los bailarines se hicieron varias pruebas simples e individualizadas por sección del cuerpo. Se pudo comprobar que cada uno de los movimientos emitían datos de forma continua.

En el caso del sensor acelerómetro, ADXL3xx, en un principio se hizo el diseño para que estuviese en el mismo cinturón donde se encuentra la caja negra, sin embargo por comodidad para el sujeto que lleve el traje y para tener una medición se optó por trasladarlo a la muñeca libre, dispersando el campo de trabajo del conjunto de sensores, obteniendo más datos en el mismo tiempo, ya que se comprobó que el tronco no tenía tanta torsión como se planteó desde un principio.

Una vez hecho los cambios pertinentes en el traje para tener una mejor y mayor movilidad, como la opción máxima de datos, se hizo una prueba final antes de la presentación a la conferencia Metabodoy [32], comprobando su perfecto funcionamiento entre sensor, lectura de datos de cada sensor, conversión en señales MIDI y la siguiente salida en forma de sonidos y luces.

Esto nos lleva a que se consiguió conectar con total precisión los movimientos del bailarín con el envío de señales MIDI, que se reciben en tiempo real con una latencia inferior a los 30 ms.

Así, como resultado del traje final se muestra en las imágenes a continuación:

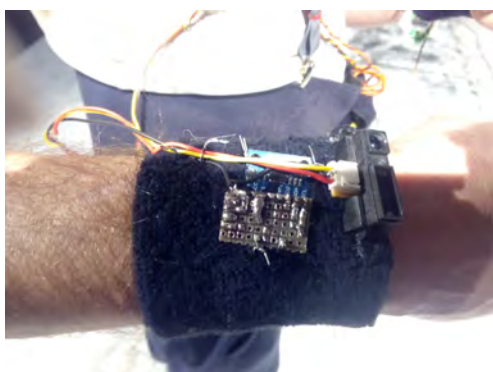


Figura 5.1: Muñequera con sensor de movimiento ADXL345



Figura 5.2: Sensor ADXL 3xx



Figura 5.3: Muñequera con sensor ADXL3xx



Figura 5.4: Bilarín con el traje completo

Capítulo 6

Conclusiones

Haciendo una valoración entre los objetivos propuestos y los resultados finales y haciendo una agrupación entre los objetivos principales, tales como el desarrollo e implementación de un traje; y secundarios como el significado que quiero transmitir a la gente a través de este proyecto, se ha de mencionar los siguientes puntos:

- Como objetivo principal el de la realización de un traje multiesensorial que sea controlador MIDI inalámbrico, se ha cumplido enteramente. Este traje cumple todos los requisitos técnicos de detección, integración de señales sensoriales y envíos de mensajes MIDI mediante el protocolo de Bluetooth.
- Una vez que se finalizó el traje, se hicieron pruebas externas para las señales MIDI a través de la aplicación Ableton LIVE.
- Como objetivo secundario, este proyecto se ha estado mostrando en varios congresos, dónde se explica la relación de cuerpo - movimiento - tecnología - arte, son el *International Metabody Laboratory*, que tuvo como cartel el que aparece en la siguiente imagen C.6 en la *Brunel university* de Londres [32] y el *Actualizando la idea de lo monstruoso Un homenaje al Frankenstein de Mary Shelley* en la Universidad Complutense de Madrid, cuyo cartel fue el que se muestra en la figura C.7 [33].

Como valoración personal, el proceso de creación de este proyecto, según mi punto de vista no ha sido extremadamente difícil, quitando el hecho de tener que aprender el funcionamiento de los sensores, y consecuentemente como poder programarlos para su correcto funcionamiento. Sin embargo me ha parecido un trabajo muy completo, he tenido que estudiar los movimientos de un bailarín, aprendiendo sus pasos para verificar que el traje no fuese un impedimento a la hora de realizar la presentación. He tenido que aprender a soldar de una forma precisa con diferentes materiales, para que el traje sea lo suficientemente resistente como flexible, y he ahondado mucho en la programación con Arduino, que hoy en día en un lenguaje muy arraigado dentro de proyectos de electrónica nivel básico, medio o avanzado.

Si que es verdad que lo que realmente hubiese hecho que hubiese sido un trabajo más complejo sería el añadir la programación con señales MIDI. Aunque cada vez el campo tecnología - arte es más extenso, y tiene más ejemplos en este tipo de programación, sigue siendo difícil el encontrar el equilibrio entre lo que se quiere transmitir y lo que se acaba transmitiendo, es decir el captar los movimientos a través de los distintos sensores se debe transmitir sin palabras, solo con efectos externos, lo que dificulta el resultado final, ya que puede variar de lo que se había planeado inicialmente.

Un dato a destacar es el hecho de haber llegado a un acuerdo entre tantas partes dispersas, especialistas en baile, especialistas en música, especialistas en el lenguaje "Processing" en tecnología - arte y yo como último eslabón que debía ser la que tenía que reunir toda la información sugerida en un único proyecto, ha llevado más tiempo por la necesidad de un estudio más minucioso en cuanto a todos los materiales a usar, como los datos que se debían leer.

Por último, como proyecto en general además de haber aprendido mucho en diferentes temas, como el arte y la tecnología, el entorno Arduino, etc. Me he sentido realizada, ya que he sido capaz de cumplir mi primer y principal objetivo, el elegir un proyecto capaz de expresar algo más que datos, el haber unificado el mundo de la electrónica, que se suele visualizar como algo frío e inexpresivo con el mundo del arte, opuesto al primero, dotando a este proyecto de un enfoque global atípico, que servirá para que gente de todo tipo pueda disfrutar en las actuaciones que se están llevando a cabo, y que espero que lo sigan haciendo en un futuro. Así mismo, quería agradecer en especial a mi tutor Javi F. Gorostiza, Alfredo Miralles y Javier Aparicio, que me han dado toda su ayuda a lo largo de todo este proceso.

Capítulo 7

Posibles mejoras y trabajos futuros

Un método de mejorar el sistema de cara al futuro es cambiar de placa microprocesadora. Esto se debe a que debido a el número limitado de pines analógicos que tiene la Arduino UNO rev 3, sólo se pudieron usar 4 sensores, lo que limita la cantidad de datos obtenidos para procesar y transformar después en efectos, y por consiguiente la transición de un sonido a otro es mucho más abrupta, a mayor puntos (correspondientes a cada movimiento) mayor fluidez en el sonido que produce el traje.

También se puede cambiar el módulo de transmisión inalámbrica, si en vez de bluetooth se utilizase un modelo Xbee - ZigBee permitiría un mayor rango en cuanto a la distancia entre el traje y el ordenador que procesa la información, obteniendo una mayor autonomía en el caso de estar en un entorno en el que no te permita tener el ordenador portátil a menos de 100 metros.

Una forma de facilitar el movimiento al bailarín sería usar hilo conductor en vvez de cables soldados, ya que con el movimiento y el roce, los cables acaban por romperse. Sin embargo con el hilo conductor, se cose sin más a una tela permitiendo mayor resistencia y movilidad para el bailarín.

Para crear mayor espectáculo además de con el movimiento y posición de los sensores, sería añadir una cámara kinect, que consta de distintos sensores, como aparece en la figura 7.1 donde reconociese el cuerpo del bailarín, pudiendo jugar con proyecciones a tiempo real, apareciendo una imagen similar a la de la figura 7.2, a través del lenguaje "Processing". Este conjunto se podría estrapolar a otros campos de enfoque, como por ejemplo el pedagógico, donde el cuerpo del usuario funcionase como una mando de control con el que puede controlar lo que pasa en la pantalla.

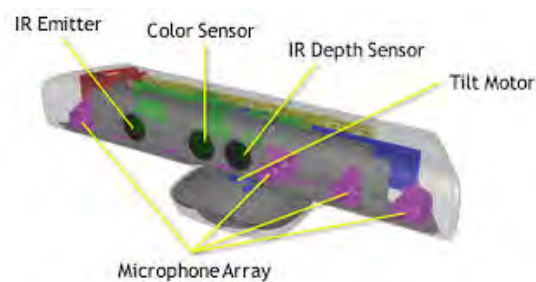


Figura 7.1: Cámara Kinect



Figura 7.2: Imágen obtenida a través de un sensor Kinect

7.1. Trabajos futuros

Cuando este proyecto empezó a tomar forma, la idea inicial era el crear un traje para una obra de teatro pedagógica. La obra en sí es *La Odisea* de Homero. Dónde simplemente con un par de objetos de atrezzo, como una mesa, un par de cortinas y una silla fuesen el entorno donde sucedería la historia.

Esta historia estaría contada a través del cuerpo del bailarín y sus movimientos, representando lo que le pasa a Odiseo, protagonista del poema, a través de las sombras creadas por la modificación de las luces y del sonido a través de la lectura de datos de los sensores que se encuentran en el cuerpo del bailarín.

Como culmen de la obra, el público podría unirse al espectáculo provándose el traje para así ver como funciona la interpolación del movimiento con el cambio de ambiente que crean con sus propios cambios. Mostrando que los nuevos avances tecnológicos se pueden encontrar en cualquier situación, y creándose así un vínculo mayor entre el espectador y la tecnología y haciendo que se quiten los prejuicios ya establecidos sobre la electrónica y la tecnología, cómo algo difícil de comprender.

APÉNDICES

Presupuesto del Proyecto

El siguiente presupuesto se ha basado según el salario base para una Ingeniero Junior y un jefe de proyecto en España, teniendo en cuenta las horas que han llevado a cabo el proyecto. Para hallar el coste, a continuación se describe un desglose de las tareas realizadas a lo largo del periodo del trabajo, que se basa en la planificación que se explicó previamente en el capítulo 1.

A.0.1. Desglose por horas

- **Fase 1: Investigación.**

Proyecto	Horas invertidas
Investigación del estado del arte	20
Conocimiento de <i>wearable tecnlogy</i>	15
Tutoriales de Arduino IDE	10
Información bibliotecas sensores	8

Tabla A.1: Desglose de horas - fase 1

■ Fase 2: Inicio del desarrollo.

Proyecto	Horas invertidas
Prueba del código IR	15
Prueba del código ADXL 3xx	5
Prueba del código ADXL 345	25
Prueba del código SWITCHES	10

Tabla A.2: Desglose de horas - fase 2

■ Fase 3: Desarrollo.

Proyecto	Horas invertidas
Prueba final IR	5
Prueba final ADXL 3xx	5
Prueba final ADXL 345	15
Prueba final SWITCHES	10
Desarrollo traje	25
Ensamblaje definitivo	35

Tabla A.3: Desglose de horas - fase 3

- Fase 4: Finalización del trabajo.

Proyecto	Horas invertidas
Prueba final traje	10
Optimización del código	5
Redacción de memoria	90

Tabla A.4: Desglose de horas - fase 4

Fases	Horas invertidas
Fase 1: Investigación.	53
Fase 2: Inicio del desarrollo.	55
Fase 3: Desarrollo.	95
Fase 4: Finalización del trabajo.	105
TOTAL	308

Tabla A.5: Desglose total de horas

A.0.2. Costes materiales

Para hacer el coste total por los materiales hay que tener en cuenta que el ordenador se considera con un periodo de amortización de 3 años. Los costes materiales han sido los siguientes:

Concepto	Precio (Euros)
Arduino UNO rev 3	20
Módulo BT EZ - Link	25,95
Sensor IR SHARP GP2D120	12,26
Sensor ADXL 3xx	13,05
Sensores ADXL 345	15,78
Muñequeras	5
Faja	10
Conectores	5,92
Cable	3,02
Láminas de Latón	8,53
Ordenador de trabajo hp Pavilion dv7	250
TOTAL	369,51

Tabla A.6: Desglose total de coste de materiales

A.0.3. Costes de personal

Personal	Horas	Precio/hora	Importe (Euros)
Jefe de proyecto	20	55	1.100
Ingeniero Junior	308	40	12.320
TOTAL	328		13.420

Tabla A.7: Costes de personal

A.0.4. Coste total final

Concepto	Precio (Euros)
Costes de personal	13.420
Costes de material	369,51
TOTAL con IVA	13.789,51

Tabla A.8: Coste total

El coste total del trabajo de fin de carrera es de :

Trece mil setecientos ochenta y nueve euros con cincuenta y un céntimos.

APÉNDICE B

Anexo I

B.1. *Wiring*

Es un entorno de programación de código abierto (*open source*) para microcontroladores. Está compuesto de un lenguaje de programación, un entorno de desarrollo integrado (IDE) y un dispositivo microcontrolador.

Wiring fue un proyecto que lo inició Hernando Barragán, junto con la ayuda de Brett Hagman y Alexander Brevig.

Permite la programación del software de una multiplataforma para controlar todos los dispositivos conectados a una variedad de microcontroladores, para crear todo tipo de códigos, espacios, objetos interactivos o experiencias físicas. El proceso de desarrollo se le llama *sketching the hardware* (dibujando el hardware), en el que mediante unos pocos comandos y la conexión de un par de componentes electrónicos se pueda observar el cambio que se quiere realizar. El entorno se creó con el propósito de fomentar una comunidad donde los principiantes en el mundo de la programación y electrónica aprendiesen a través de aquellos usuarios más experimentados, compartiendo ideas, diseños, conocimientos y experiencias. Este entorno está basado a su vez en *Processing*.

B.2. *Processing*

El procesamiento es un entorno y lenguaje de programación de desarrollo integrado de texto fuente basado en Java, aunque puede utilizarse otros lenguajes de programación similares, que se utiliza para el diseño digital. La fundación fue creada por Ben Fry y Casey Reas.

El programa crea los llamados bocetos textit sketches, que se almacenan en el textit sketchbook. Aquí se pueden dibujar gráficos de dos y tres dimensiones, dependiendo del procesador que se use. Dentro del textit processing nos encontramos con una cantidad de librerías que poseen cada una de sus características enfocada a un tema. Las capacidades de procesamiento se amplían con estas bibliotecas y herramientas. También tiene diferentes modelos de programación, para que los bocetos puedan desplegarse en diferentes plataformas y programas de diferentes maneras.

Anexo II

A continuación mostrare algunas de la imagenes recopiladas durante la primera presentación del conjunto del trabajo SynBody.



Figura C.1: Actuación durante una de las presentaciones



Figura C.2: Bailarines durante la actuación

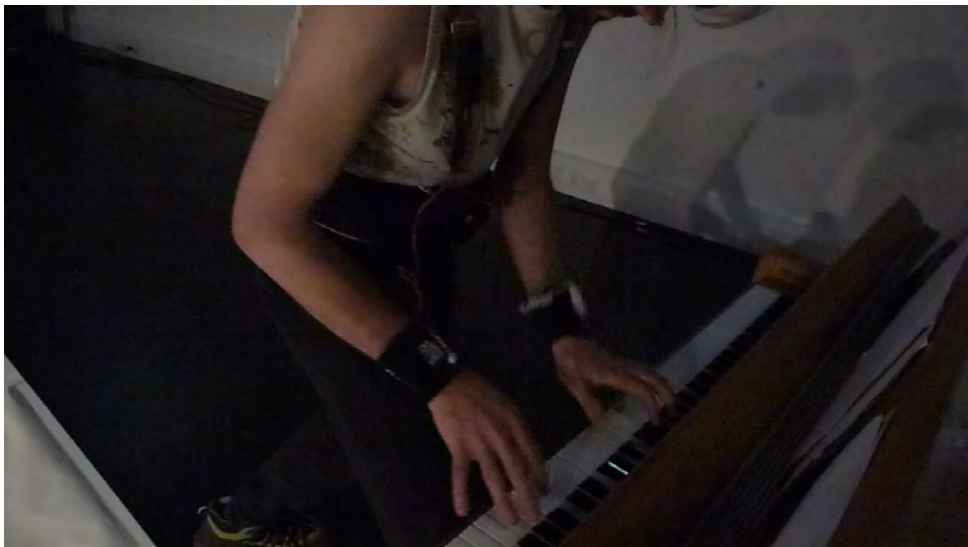


Figura C.3: Ejemplo de distintas funciones del trabajo



Figura C.4: Traje interactuando con el público



Figura C.5: Parte de la actuación del proyecto SynBody

Imagen del cartel del Forum Internacional de Metabody:



Figura C.6: Cartel Metabody

Imagen del cartel de la presentación en Bellas Artes:



Figura C.7: Cartel de Bellas Artes

Bibliografía

- [1] Robert Faludi. *Buiding Wireless Sensor Networks*. O'Reilly, Sebastopol, California, 2010.
- [2] Federico Lorussi, Enzo Pasquale Scilingo, Mario Tesconi, lessandro Tognetti, and Danilo De Rossi. *Strain Sensing Fabric for Hand Posture and Gesture Monitoring*. IEEE TRANSACTION ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL 9. NO.3, 2005.
- [3] Chulsung Park and Pai H. Chou. *A Wearable Wireless Sensor Platform for Interactive Dance Performance*. IEEE Pervasive Computing and Communications, 2006.
- [4] Wiring Challenges 3: Musical Controllers,
[http : //wiki.wiring.co/wiki/Wiring_Challenges3 :_Musical_Controller/es](http://wiki.wiring.co/wiki/Wiring_Challenges3:_Musical_Controller/es).
- [5] José Manuel Ruiz Gutiérrez. *Implementación de Sistemas de Transmisión de Datos y Sensores en Redes Inalámbricas con XBee Integrado en la "Plataforma Open Hardware.Arduino*. Arduino Comunicación, 2012.
- [6] Alexandros Kitsikidis, Kosmas Dimitropoulos, Stella Douka, and Nikos Grammalidis. *Dance Analysis using Multiple Kinect Sensors*. Informatics and telematics Institute, 2013.
- [7] Aylward, R. *Senseable: A Wireless Inertial Sensor System for Interactive Dance and Collective Motion Analysis*. Masters of Science in Media Arts and Sciences, Massachusetts Institute of Technology, 2006.
- [8] Adafruit: Introducing Bluefruit EZ-Link Breakout ,
<https://learn.adafruit.com/introducing-bluefruit-ez-link/tour>

-
- [9] Arduino: Button,
<https://www.arduino.cc/en/Tutorial/Button>
 - [10] IR Distance Sensor and Push Button,
<http://www.robotshop.com/blog/en/arduino-5-minute-tutorials-lesson-4-ir-distance-sensor-push-button-2-3637>
 - [11] Three Fabric Buttons,
<http://www.instructables.com/id/Three-Fabric-Buttons/?ALLSTEPS>
 - [12] Tap, Tap, Drop. ADXL345Accelerometer + Arduino,
<http://bildr.org/2011/03/adxl345-arduino/>
 - [13] ADXL345 Quickstart Guide,
<https://www.sparkfun.com/tutorials/240>
 - [14] ADXL3xx Acccelerometer,
<https://www.arduino.cc/en/Tutorial/ADXL3xx>
 - [15] Aula de las Artes UC3M,
<https://auladelasartes.uc3m.es/>
 - [16] J. A. Paradiso and F. Sparacino, *Optical Tracking for Music and Dance Performance*. Optical 3D Measurement Technics, 1997.
 - [17] Cunningham, bailarín,
<http://www.danza.es/multimedia/biografias/merce-cunningham>
 - [18] Alex Posada, artista,
<http://www.artsmoved.cat/en/alex-posada/>
 - [19] Checks on Speed, grupo musical contemporáneo,
<http://www.chicksonspeed.com/>
 - [20] MIDI Controllers,
<https://ccrma.stanford.edu/craig/articles/linuxmidi/misc/essenmidi.html>
 - [21] MIDI Essentials,
<https://www.midi.org/specifications/item/table-1-summary-of-midi-message>

-
- [22] Danza y proyecciones,
<http://dartecne.wikidot.com/dance-and-projections>
- [23] Jorge Sad de Levi, compositor,
<https://jorgesadlevi.wordpress.com/bio/>
- [24] Silvina Szperling, coreógrafa,
<http://www.alternativateatral.com/persona6375-silvina-szperling>
- [25] Garry Hills, artista,
<http://dig.henryart.org/northwest-artists/artist/gary-hill>
- [26] Tamara Cubas, artista,
http://www.keyperformance.se/?page_id=45
- [27] R. Morales - Manzanares, *SICIB: Sistema Interactivo de Composición e Improvisación Musical para Bailarines*. Laboratorio de Informática Musical.
- [28] *The nutcracker*, en el ballet deBrookling,
<http://brooklynballet.org/performances/brooklyn-nutcracker>
- [29] Danza y tecnología,
<http://danzatecnologia.blogspot.com.es/?view=classic>
- [30] Dust by Robert Ashley,
<http://www.princemyshkins.com/dust.pdf>
- [31] Javier Aparicio, Proyecto: SynBody,
<http://alfabody.net/en/projects/synbody/>
- [32] Metabody,
<http://metabody.eu/imf-2016-london/>
- [33] Universidad Complutense Bellas Artes - Actualizando la idea de lo monstruoso,
<http://bellasartes.ucm.es/actualizando-la-idea-de-lo-monstruoso>